Feature Preserving Image Translation for Gaze Estimation

Marcel Bühler

Semester Project
March 2019

Supervisors:
Seonwook Park
Dr. Xucong Zhang
Prof. Dr. Otmar Hilliges

ETH zürich

Advanced Interactive Technologies

Abstract

Collecting labelled data for convolutional neural networks is a very expensive and time-consuming process. In the domain of eye gaze estimation, it is possible to generate labelled synthetic images using graphical renderers. We explore a promising approach to image style transfer via Generative Adversarial Networks (GAN). The aim is to reduce the domain gap between synthetic and real-world eye images. In this work, we combine recent developments in the field, such as Deep Convolutional GAN (DCGAN) or CycleGAN, with custom loss functions as suggested by Sela et al. and Lee et al.

The main goal of this work is to investigate the effect of feature preservation for unpaired image translation. Specifically, we re-implement the architecture proposed by Lee et al. with the extension of a feature preserving cost function. We go a step further by preserving eye gaze directions, as well as regional landmarks when training the image translation networks. As a qualitative result, the generated images in the fake domains look very similar to images in the original domains. Quantitatively, we show a clear improvement in eye gaze prediction accuracy when putting higher weights on the feature-preservation.

Contents

List of Figures

List of Tables

1

Introduction

Human eye gaze can be used in many human-computer interaction systems. The users' eye gaze can directly reflect their visual attention, which can be used as explicit input for the system [Kurauchi et al. 2016]. The long-term gaze pattern can be applied for user modeling for behaviour analysis [Sattar et al. 2015]. It can also be an important element in safety, for example when observing a loss of attention for train drivers or airplane pilots. Modern cars also rely on eye tracking in order to learn about the driver's state of attention.

Eyes can enable people to operate a computer without using a mouse. It is possible to type [Mott et al. 2017] or choose items [Zhang et al. 2014] via eye gaze tracking. Besides that, the eye gaze gives considerable information about a person's current state of mind. For example, it can reveal the user's interest, engagement, attention and emotional state [Lagun et al. 2014, Li et al. 2017, Faber et al. 2017]. It can also tell a lot about the relationship between two humans. In a study, [Bolmont et al. 2014] found a relation between eye movement and sexual desire. Furthermore, the eyes reveal information about the mental health status. [Hutton et al. 1984] related eye tracking dysfunction to Alzheimer dementia. Other work found relations to Parkinson and schizophrenia [Kuechenmeister et al. 1977, Kühborth 2017].

Given all the information that can be learned from eye gaze tracking, it would not come as no surprise if tracking devices soon became more and more prevalent in daily life. Most personal devices, such as laptops, phones or tablets include a front-facing camera and can therefore be used for eye tracking.

As a matter of fact, real-life eye gaze tracking is not an easy task. Traditionally, eye gaze is estimated via model-based or feature-based approaches, where an algorithm first identifies facial and regional landmarks, e.g. the iris and pupil position, and then estimates eye gaze based on these features [Zhou et al. 2017, Hansen and Ji 2010]. More modern approaches, such as appearance-based eye gaze estimators, have become more robust than model-based approaches

and tend to perform better in a broader range of unconstrained settings. However, there are some technical challenges. Appearance based methods require a labelled set of training data, whose collection is expensive and time-consuming. The dataset needs to be of sufficient variability and the collection procedure needs to be carefully defined to yield consistent labels. The collected images might be recorded by mobile cameras and therefore suffer from noise, low resolution or a bad illumination. There might also be a large variance in head pose. In particular the currently best performing approaches via deep learning require large amounts of labelled data.

Arguably the most challenging benchmark for evaluating gaze estimation methods is cross-dataset gaze estimation. Newer appearance-based methods have progressively improved on this benchmark in particular on the MPIIGaze dataset, improving from $13.9°$ [Zhang et al. 2015] to $10.8°$ [Zhang et al. 2017b]. However, these methods are unable to out-perform model-fitting methods ($8.3°$ [Park et al. 2018]) or a simple k-NN method on synthetic data ($9.95°$ [Wood et al. 2016]) which make use of larger amounts of annotated training data.

A solution to the lack of annotated quality data can be the use of synthetic images. For eye gaze estimation, there exist tools that are able to create an unlimited set of labelled images [Wood et al. 2016]. Unfortunately, these synthetic images do not follow the same appearance as real world images. They are not realistic enough to be directly applied for training highly accurate eye gaze estimation models. The gap between the look of synthetic and real images can be reduced by a function that maps images from the synthetic to the real domain. Such a function can be learned by training Generative Adversarial Networks (GANs) [Goodfellow et al. 2014]. There have been several approaches applying GANs to image generation and translation [Radford et al. 2015, Zhu et al. 2017].

This work was inspired by previous work in unpaired image-to-image translation in the field of eye gaze estimation [Shrivastava et al. 2016, Lee et al. 2018, Sela et al. 2017]. In particular, [Shrivastava et al. 2016] report a large improvement in cross-dataset gaze estimation error of $7.8°$ and follow-up works have yet to demonstrate further improvements. We observe that so far, these approaches do not specifically preserve image features that are relevant to eye gaze estimation. Instead, they apply a rather generic loss term that is based on the L1 loss of pixel values.

In this work, we re-implemented the architecture from [Lee et al. 2018] and extended it with a feature preserving cost function. The goal was to specifically preserve the eye gaze direction and regional landmarks in the image translation process. This was achieved by estimating the eye gaze direction and regional landmarks location on both the original and the produced images and punish the generator in case of a discrepancy. The generator learned to generate images that preserve these image features. In order to balance the trade-off between the various feature losses, we introduce weighting constants to the individual terms of the cost function. By experimenting with different combination of weighting constants, we could find values that yielded the best quantitative results for eye gaze estimation. We found a clear correlation between the feature weights and the model accuracy. Higher values for feature weights yield better model performance. We also qualitatively compared the output images generated by different GANs trained with different feature weights. The outputs underline the findings from the quantitative evaluation.

1.1. Contributions

This work explores the influence of feature preserving model architectures for unpaired image translation. It makes the following contributions:

1. Re-implementation of the architecture suggested by [Lee et al. 2018] in tensorflow [Abadi et al. 2015].
2. Direct application of eye gaze consistency.
3. Exploration of eye shape consistency through regional landmarks preservation.

2

Related Work

2.1. Eye Gaze Estimation

There are two main approaches to eye gaze estimation: model-based and appearance-based eye gaze estimation. The model-based methods first detect regional landmarks, such as the pupil, iris and eye corners and use the location of these landmarks for predicting the eye gaze. Appearance-based methods estimate eye gaze directly from eye images. Early models were trained on small datasets and did not generalise well [Lu et al. 2011]. By now, appearance-based methods have started to outperform model-based approaches. In particular in noisy settings, these methods tend to be more robust than model-based techniques. As a drawback, appearance-based models require large amounts of labelled training data, which are expensive to acquire. The most popular real-world datasets are MPIIGaze [Zhang et al. 2017b] with 213659 images from 15 people, GazeCapture [Krafka et al. 2016] with 2.5 million images from 1450 people and TabletGaze with 100000 pictures from 35 people [Huang et al. 2015]. In this work, we use a subset of the MPIIGaze (MPIIFaceGaze [Zhang et al. 2017a] with 37639 images from 15 people) as a real-world dataset.

The prediction accuracy of eye gaze prediction has significantly improved in the recent years, in particular for the cross-dataset setting, arguably the most challenging benchmark. The angular error on the MPIIGaze dataset dropped from over 16 degrees in 2015 [Zhang et al. 2015, Lu et al. 2014] to less than 8 degrees [Shrivastava et al. 2016, Lee et al. 2018]. The reasons for this improvement is the availability of tools to generate synthetic images [Wood et al. 2016], as well as advances in deep learning. GANs can be used to refine synthetic images in order to make them look more realistic and achieve an accuracy below 8 degrees. In the following, we describe the synthetic dataset and related work in the field of deep learning.

2.2. Synthetic Dataset

Training deep appearance based eye gaze prediction models requires a large amount of labelled training data of sufficient variability. [Wood et al. 2016] developed *UnityEyes*, a tool for synthesising eye images, with the goal to create a larger training set without the need of expensively collecting training images. *UnityEyes* allows the fast rendering of synthetic eye images including annotations for eye gaze, head pose and regional landmarks. The method uses a 3D model of the eye region and is capable of synthesising a large quantity of eye images in a short time. The authors showed state-of-the-art cross-dataset performance for eye gaze estimation on real images from the MPIIGaze dataset [Zhang et al. 2017b] using a simple k-nearest-neighbour algorithm. They stated that training machine learning models on synthetic data had the potential to reduce the need of expensively collecting and labelling data. However, the synthetic images do not follow the exact same distribution as real images. Aligning these distributions by learning a function that translates images from one domain to the other can further improve the model accuracy. [Shrivastava et al. 2016] proposed to learn this function via GANs. Their approach was called *Simulated+Unsupervised* (S+U) *learning*. They trained a GAN, which mapped synthetic images to the real domain, a process called *image refinement* or *image translation*. This allowed the creation of an arbitrarily large training set of produced real images. Their resulting models showed considerable improvements compared to models trained on purely synthetic data. They also suggested to update the discriminator using a history of refined images in order to stabilise training.

In this work, we work with images generated by UnityEyes and we update the discriminator using a history of refined images.

2.3. Generative Adversarial Networks and Style Transfer

Generative Adversarial Networks (GANs) [Goodfellow et al. 2014] have become more and more popular for learning feature representations in image classification tasks. They are also often used for image generation. [Radford et al. 2015] successfully combined GANs and Convolutional Neural Networks (CNNs) to unsupervised learning tasks. They called this family of models *Deep Convolutional Generative Adversarial Networks* (DCGANs). DCGANs follow architectural guidelines that stabilise the network training. As an example, they applied batch normalisation and used (leaky) ReLU activation functions for most layers. They also recommended to use (fractionally) strided convolutions instead of deterministic pooling layers. The authors demonstrated that DCGANs learn specific object representations, such as windows. They also demonstrated that simple arithmetic operations on the input space (e.g. addition and subtraction) lead to rich linear structure in the representation space. In our work, we follow the principles of DCGAN.

Another important topic in representation learning is image style transfer. [Gatys et al. 2016] addressed the problem of explicitly expressing the semantic information in an image. They introduced a *Neural Algorithm of Artistic Style*, which could differentiate between semantic image content and image style. This was achieved by optimising a loss function consisting

of separate loss terms for content (the values for the weights) and style (feature correlations represented by the Gram matrix of feature weights).

In practice, unpaired image-to-image translation using deep neural networks is a highly under-constrained problem and the output of a standard GAN is often not satisfactory without careful tweaking of hyper-parameters. In order to improve the quality of the output images, as well as enforcing a better preservation of the input, [Zhu et al. 2017] introduced the *cycle-consistency loss*. In contrast to other unpaired image-to-image translation methods, such as variational auto-encoders [Kingma and Welling 2013, Rezende et al. 2014], they did not assume that an image needed to be embedded in a low dimensional space or that the image style was represented by the Gram matrix of its feature weights [Gatys et al. 2016]. They also further reduced the number of parameters in the discriminator by classifying patches of the output images (PatchGAN [Isola et al. 2016]). As an additional benefit, this also allowed predictions on arbitrarily-sized images. Furthermore, they kept a history of generated images for updating the discriminator, which stabilised the training process.

The approach by [Shrivastava et al. 2016] was extended by [Sela et al. 2017] and [Lee et al. 2018]. Sela et al. built a model for eye gaze estimation on mobile devices (GazeGAN). The GazeGAN model was trained on produced real images (created by a CycleGAN). To preserve the eye gaze directions in the produced real images, they extended their cost function by adding a loss term for *gaze cycle-consistency*. As a result, they demonstrated highly accurate eye gaze predictions under most settings. However, the translation from the synthetic to the real domain occasionally failed in the case of extreme head poses or peculiar skin textures. Similarly, [Lee et al. 2018] proposed an end-to-end S+U learning algorithm that preserved features in the translation process. Their approach used the CycleGAN loss function with the addition of a general feature preservation loss term (*feature-consistency loss*). This loss encouraged the model to preserve image features in the translation process. In their paper, Lee et al. used the identity map as feature extractor, but other feature extractors would be possible. Unlike Sela at al., they did not only make predictions on test images in the real domain, but in addition, they also translated real test images from the MPIIGaze dataset into the synthetic domain and used a model trained on purely synthetic data. Their eye gaze prediction model outperformed previous approaches on the MPIIGaze dataset [Shrivastava et al. 2016, Zhang et al. 2017b]. Our model very closely follows the architecture from [Lee et al. 2018].

3

FP-GAN: Feature Preserving Generative Adversarial Network

The focus of this project is to experimentally investigate the effects of different factors on the unpaired image translation and gaze estimation performance. Fig. 3.2 gives an overview of the model components. In the following, we describe the architecture, cost function, training procedure and discuss our results.

3.1. Cost Function

The standard loss for a GAN as introduced by [Goodfellow et al. 2014] depends on the output of the Discriminator D, given the real input $\boldsymbol{x}$ and the fake input $G(\boldsymbol{z})$. The fake input $\boldsymbol{z}$ is usually a noise vector.

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[log(D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[log(1 - D(G(\boldsymbol{z})))] \tag{3.1}$$

Building on top of this, [Zhu et al. 2017] developed *CycleGAN*. CycleGAN follows [Goodfellow et al. 2014], but applies an adversarial loss in both directions ($\boldsymbol{x}$ to $\boldsymbol{z}$ and $\boldsymbol{z}$ to $\boldsymbol{x}$). There are two generators G and F, both with an own discriminator D_G and D_F. Each pair follows the loss function above (3.1). In our method, called **F**eature **P**reserving Generative Adversarial Network (FP-GAN), we extend the CycleGAN loss function by adding feature-specific loss terms.

We use the following notation. Let S denote the synthetic domain represented by image generated by UnityEyes [Wood et al. 2016]. When we refine an image in S, we translate it to

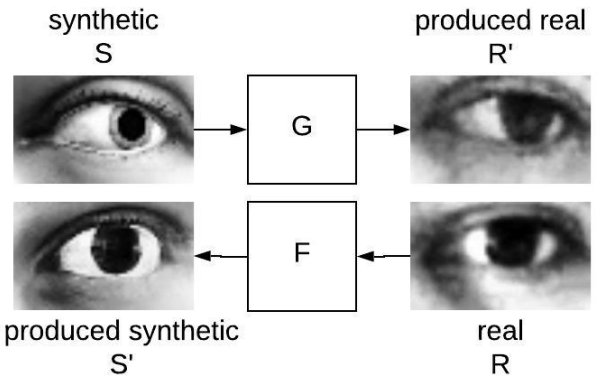


Figure 3.1.: *Overview of domains and notation. On the left, we have the synthetic domain S and the produced synthetic domain S'. On the right, we have the real domain R and the produced real domain R'. The generator G translates images from the synthetic domain S to the produced real domain R'. F is its counterpart for the other direction.*

the produced real domain R'. The other direction follows analogously. Images from the real domain R are translated into the produced synthetic domain S'. Fig. 3.1 shows the relation between the domains and the two generators. The first generator G translates images from S to R' and the second generator F takes images from R and maps them to S'.

We apply three feature-specific loss terms. The first one, *identity transform loss* ($\mathbb{L}_{id}$), punishes changes in single pixel values. The second one, *eye gaze consistency loss* ($\mathbb{L}_{eg}$) penalises shifts in eye gaze direction. The third term, *landmark consistency loss* ($\mathbb{L}_{lm}$), preserves the shape of the eyelid and the iris. Fig. 3.3 illustrates the model architecture for calculating the eye gaze and landmarks feature losses. The generator takes an original image as input and translates it to the refined domain. Then, a previously trained gaze estimator, whose weights are fixed, predicts the eye gaze direction on both the original and the produced image. The loss is calculated as the mean L2 distance between the predictions. The process for calculating the landmarks loss is the same, but it uses a landmarks detector instead of a gaze estimator. As we are using a CycleGAN and therefore translate in both directions, this process happens in both directions.

Fig. 3.4 shows the overview of all losses in our generative model. We translate images from one source domain X (e.g. the synthetic domain S) to the target domain Y (e.g. the produced real domain R'). Then, we estimate the feature specific losses for eye gaze, landmarks and identity-transform between X and Y, as well as the cycle-consistency loss between X and X'.

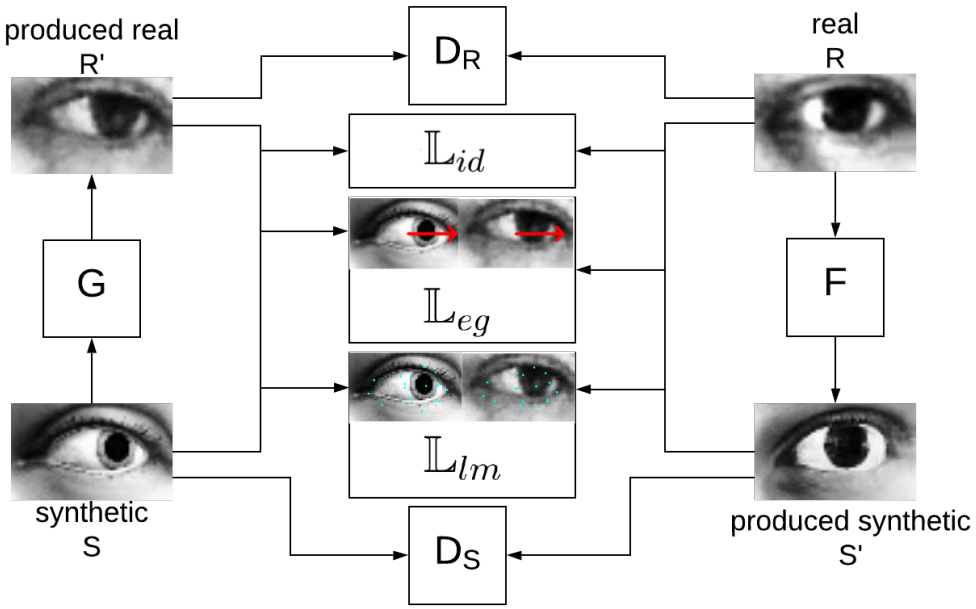
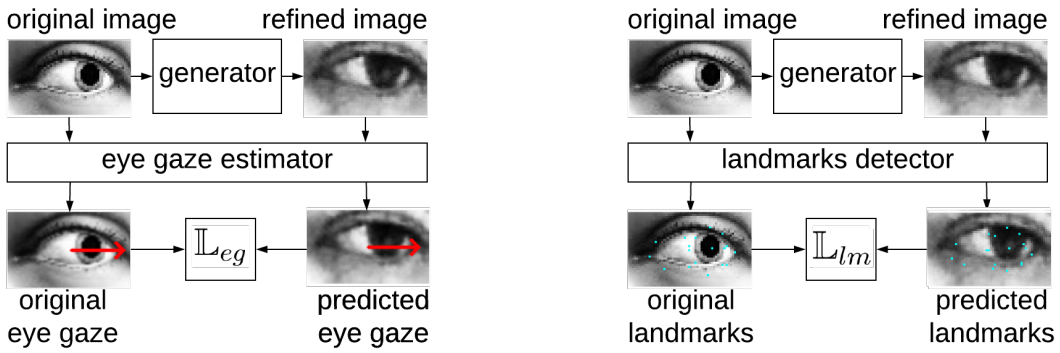


Figure 3.2.: *Architectural overview of the FP-GAN components. There are four domains, two of them represent the original images from MPIIGaze (R) and UnityEyes (S). The other two domains, R' and S', represent images produced by the generators G and F. The standard GAN loss from the discriminators (D_S and D_R) is extended with a cycle loss and a feature loss. For the eye gaze consistency ($\mathbb{L}_{eg}$) and landmarks losses ($\mathbb{L}_{lm}$), we trained two additional Convolutional Neural Networks. The identity-transform loss ($\mathbb{L}_{id}$) can directly be calculated. The cycle-consistency loss ($\mathbb{L}_{cyc}$) is being omitted in order to reduce cluttering. For an illustration, please refer to Fig. 3 of the original CycleGAN paper [Zhu et al. 2017].*



(a) (b)

Figure 3.3.: *Model architecture for eye gaze and landmarks feature losses. The original image is translated from the source domain (here the synthetic domain S) to the target domain (here the produced real domain R'). Feature extraction models predict eye gaze and landmarks. The loss is the averaged L2 distance between the feature estimates. The same architecture applies for the other direction, i.e. when translating from the real domain R to the produced synthetic domain S'.* (a) *The eye gaze estimator predicts eye gaze directions.* (b) *The landmark detector yields the coordinates of* 16 *regional landmarks.*

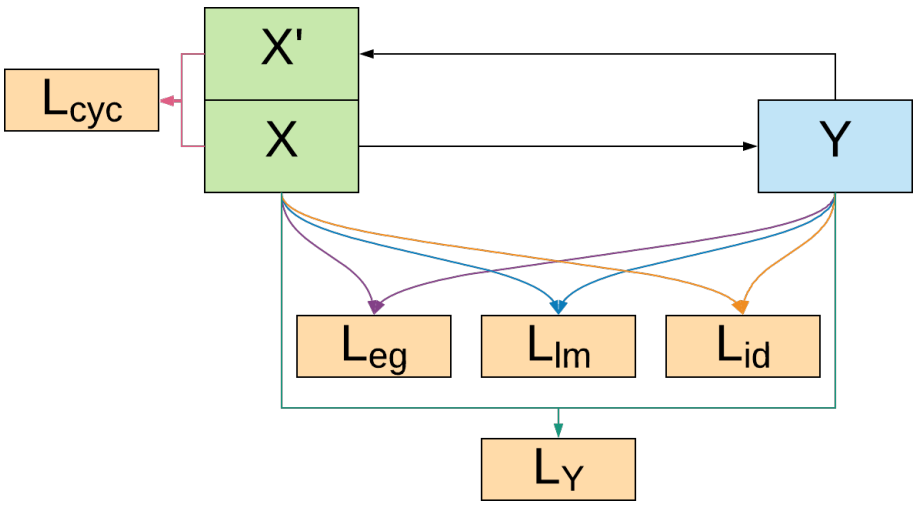


Figure 3.4.: *The loss terms of the FP-GAN when translating from an input domain X (e.g. the real domain R) to an output domain Y (e.g. the produced synthetic domain S'). $\mathbb{L}_Y$ is the loss from the discriminator and $\mathbb{L}_{cyc}$ the cycle-consistency loss. The eye gaze loss $\mathbb{L}_{eg}$ should preserve the eye gaze direction, the landmarks loss $\mathbb{L}_{lm}$ punishes the displacement of regional landmarks and the identity transform loss $\mathbb{L}_{id}$ keeps pixel values.*

The full cost function for FP-GAN is as follows:

$$\begin{aligned}\mathbb{L} =& \mathbb{L}_G + \mathbb{L}_F \\ &+ \lambda_{cyc}\mathbb{L}_{cyc} \\ &+ \lambda_{id}\mathbb{L}_{id} \\ &+ \lambda_{eg}\mathbb{L}_{eg} \\ &+ \lambda_{lm}\mathbb{L}_{lm}\end{aligned}$$

where $\mathbb{L}_G$ and $\mathbb{L}_F$ are the standard GAN losses in both directions (see eq. 3.1) and $\lambda_{cyc}, \lambda_{id}, \lambda_{eg}, \lambda_{lm}$ are weighting constants for the corresponding losses. The specific loss terms are defined as follows:

$$\begin{aligned}\mathbb{L}_{cyc} =& \mathbb{E}_S||F(G(S)) - S||_1 && + \mathbb{E}_R||G(F(R)) - R||_1 \\ \mathbb{L}_{id} =& \mathbb{E}_S||G(S) - S||_1 && + \mathbb{E}_R||F(R) - R||_1 \\ \mathbb{L}_{eg} =& \mathbb{E}_S||eg(G(S)) - eg(S)||_2 && + \mathbb{E}_R||eg(F(R)) - eg(R)||_2 \\ \mathbb{L}_{lm} =& \mathbb{E}_S||lm(G(S)) - lm(S)||_2 && + \mathbb{E}_R||lm(F(R)) - lm(R)||_2\end{aligned}$$

The estimates for eye gaze, $eg(X) \in \mathbb{R}^2$, and landmarks, $lm(X) \in \mathbb{R}^{16}$, are given by Convolutional Neural Networks pre-trained on synthetic data. The weights for these networks are kept fixed during GAN training. Please refer to the following section for more details.

For the cycle consistency loss, we use the same value as [Zhu et al. 2017] ($\lambda_{cyc} = 10$) for all our experiments. The values for the feature weights, $\lambda_{id}, \lambda_{eg}, \lambda_{lm}$, and their effect on the translations are reported below.

3.2. Feature Extraction Models

We trained convolutional neural networks (CNN) in order to calculate the feature losses for eye gaze ($\mathbb{L}_{eg}$) and eye region landmarks ($\mathbb{L}_{lm}$). In the following section, we describe the architecture and parameters used for training these models.

3.2.1. Eye Gaze Prediction Model

The eye gaze prediction model very closely followed the one from [Shrivastava et al. 2016]. The only exception is that we did not apply an L2 regularisation in the last layer. The network was trained on 100000 synthetic images from UnityEyes [Wood et al. 2016] with data augmentation (blurring, rotation and scaling). The input, 72×120 RGB or 36×60 gray-scale images, were propagated through five convolutional layers, two max-pooling layers and three fully connected layers. The weights were initialised using Xavier [Glorot and Bengio 2010] for the convolutional layers and a normal distribution for all other layers (std $= 0.02$). The last layer outputs a two-dimensional eye gaze vector encoding pitch and yaw in radians. Here is an overview of all layers for the eye gaze estimation model:

1. Convolutional layer with 32 filters and stride 2
2. Convolutional layer with 32 filters and stride 1
3. Convolutional layer with 64 filters and stride 1
4. Max-pooling layer (3×3) and stride 2
5. Convolutional layer with 80 filters and stride 1
6. Convolutional layer with 192 filters and stride 1
7. Max-pooling layer (2×2) and stride 2
8. Fully connected layer with 9600 neurons
9. Fully connected layer with 1000 neurons
10. Fully connected layer with 2 neurons

All convolutional layers first applied batch normalisation and then used leaky ReLU with a slope of 0.2 as activation function. As an optimiser, we used Adam (first moment decay rate 0.5 and second moment decay rate 0.99) and trained the network for 100000 steps with a batch size of 128 (128 epochs) and a learning rate of 2×10^{-4}. As a loss function, we applied mean squared error. We tried versions with and without L2 regularisation of parameters and found the network to perform better without any additional regularisation besides batch norm.

3.2.2. Landmark Detection Model

For the landmark detection, we applied a model trained on purely synthetic data from 1 million UnityEyes [Wood et al. 2016] as described by [Park et al. 2018]. The model was an adaption of the original hourglass architecture [Newell et al. 2016] with three hourglass modules. It was trained for 41.4 epochs (batch size 32).

3.3. Model Architecture and Training Configuration

The architecture of the FP-GAN generators and discriminators broadly followed [Zhu et al. 2017], who in turn had built their model based on the architecture suggested by [Johnson et al. 2016]. As a modification to the original model, we added custom loss terms as described in section 3.4.

The CycleGAN generator applied residual blocks. The number of blocks depended on the image size. In our case, the input images were rather small, so we use six residual blocks. The discriminators follow the PatchGAN from [Isola et al. 2016].

We experimented with a broad range of settings, focusing on using different inputs and weights for the feature loss terms. We started by training on 120×72 pixel RGB images, but soon reduced the input to 60×36 images in order to iterate more quickly. The aim was to see the relative improvements for different weights for the loss terms and not to achieve a model with

the absolute best performance.

In the following, we give an overview of the settings that we applied for our experiments.

3.3.1. Basic FP-GAN

The first version was based on RGB images of size 120×72. We started off with $\lambda_{id} = \lambda_{eg} = \lambda_{lm} = 0$, but the model did not converge. We then retrained with $\lambda_{eg} = \lambda_{lm} = 0$ and $\lambda_{id} > 0$ and the model converged. The loss function therefore was $\mathbb{L} = \mathbb{L}_{\mathbb{G}} + \mathbb{L}_{\mathbb{F}} + \lambda_{cyc}\mathbb{L}_{cyc} + \lambda_{id}\mathbb{L}_{id}$. We train the FP-GAN with λ_{id} between 1 and 15 and experimentally choose $\lambda_{id} = 2$.

3.3.2. Simplistic FP-GAN

For the simplistic GAN, we reduced the image size to 60×36 and only considered gray-scale images. This allowed faster training and hence, a higher frequency of experimental iterations. As for the Basic FP-GAN, we did not consider losses for eye gaze or landmarks. We also kept $\lambda_{id} = 2$.

3.3.3. Eye Gaze FP-GAN

The Eye Gaze FP-GAN was built on top of the Simplistic FP-GAN. As input, it took 60×36 gray-scale images. Here, we included the loss term for eye gaze $\mathbb{L}_{eg}$. We experimented with λ_{eg} between 1 and 50 and λ_{id} between 0 and 1.

3.3.4. Landmarks FP-GAN

The Landmarks FP-GAN was also based on the Simplistic FP-GAN. It was fed 60×36 gray-scale images. As loss terms, we applied the (x, y) locations of 16 regional landmarks $\mathbb{L}_{lm}$. We ran our experiments with λ_{lm} ranging from 1 to 100 and λ_{id} from 0 to 10.

3.3.5. Training

The FP-GAN variants were trained on 100000 augmented images from UnityEyes [Wood et al. 2016] and 37639 images from the MPIIFaceGaze dataset [Zhang et al. 2017a]. All versions of FP-GAN were trained for 150000 steps with a batch size of 8. We used instance normalisation [Ulyanov et al. 2016]. As an optimiser, we applied Adam [Kingma and Ba 2014] with a first moment decay rate of 0.5 and a second moment decay rate of 0.99. Following [Zhu et al. 2017], we started with a learning rate of 2×10^{-4} and linearly decayed the rate to zero starting after half of the steps. In order to improve training stability, we replaced the negative log likelihood in $\mathbb{L}_G$ and $\mathbb{L}_F$ by a least-squares loss as suggested by [Mao et al. 2016].

Following [Shrivastava et al. 2016], we updated the discriminators with a history of 50 images.

3.4. Results

We evaluated both quantitatively and qualitatively. For the quantitative evaluation, we trained eye gaze estimation models on real and fake data and compared their accuracy. For the qualitative evaluation, we visually compared the outputs of the generators to the input and to other images in the target domain. The results are reported in the following sections.

3.4.1. Quantitative Evaluation

For the quantitative evaluation, we compared the accuracy of eye gaze estimation models trained on fake and real data. First, we established a baseline by training an eye gaze estimation model $M_{Baseline}$ on 80000 augmented synthetic images from UnityEyes.

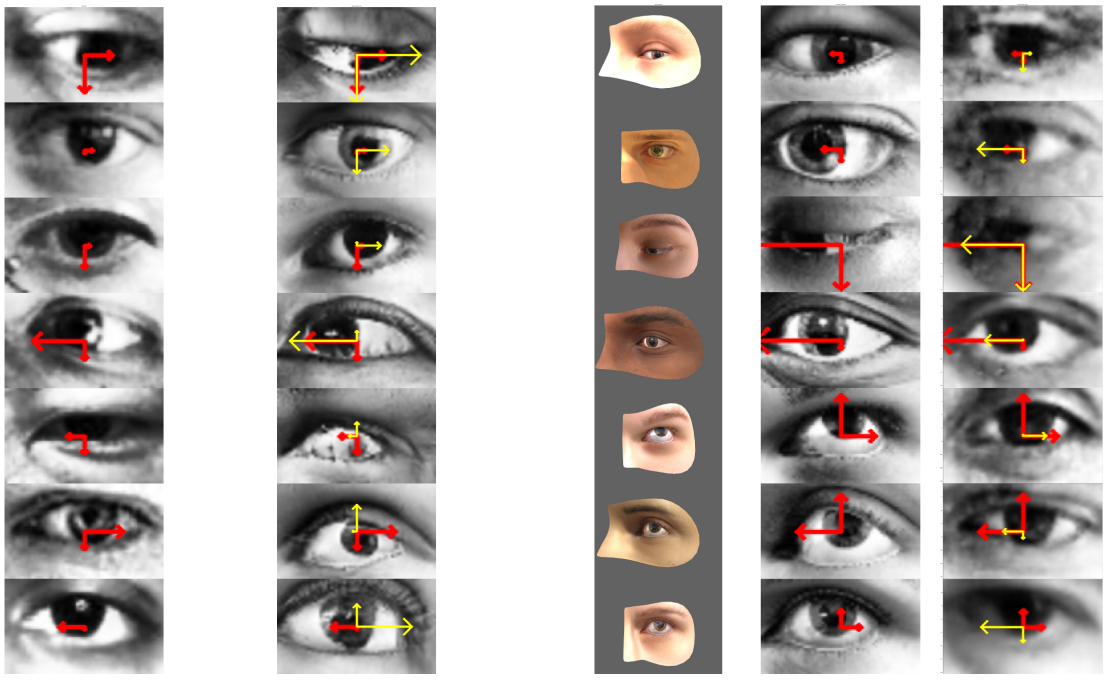
We tested on 10000 synthetic images from UnityEyes, the domain S, and on the full MPIIGaze dataset (37639 real images), the domain R. Testing on S yielded a mean angular error of 3.3621 and the error on R yielded 9.8988.

We trained a range of FP-GANs with different values for λ_{id}, λ_{eg} and λ_{lm}. For each GAN, we translated the synthetic images S to the domain R' and the real images R to the domain S'. We first evaluated the accuracy of the eye gaze estimation on produced synthetic images (S') using the baseline model, which had been trained on purely synthetic data in S. For the best-performing models (the models with the lowest angular error), we conducted further evaluation by training an eye gaze estimation model on produced real images in R' and directly tested on real images from R. These two evaluations can be called *between* because we train and test in different domains. Table 3.1 displays the results for the first *between* evaluation where we translated the test images from R to S' and estimated eye gaze on the produced images. The models using a high λ_{eg} yield the best results. The low performance for the models applying the landmarks loss $\lambda_{lm} > 0$ are explained in the discussion section. The models exclusively applying the identity-transform loss ($\lambda_{id} > 0$ and $\lambda_{eg} = \lambda_{lm} = 0$) only converged for $0 < \lambda_{id} < 5$. In the other cases, the discriminator losses went to zero while the generator losses increased.

In addition to this, we also performed a *within* evaluation by training and testing in the produced synthetic domain S'. The split was defined by taking produced images from the first 12 people (33040 images in total) in the MPIIGaze dataset for training and images from the remaining 3 people for testing (4560 test images).

Table 3.2 shows the results for all evaluation types (between and within). The scores indicate the mean angular error for the given setting. The baseline score for a model trained on synthetic data (not applying the FP-GAN) is 9.8988 for R2S and 5.6658 for within. Obviously, the image refinement does not benefit the eye gaze estimation. Please refer to the discussion section for and explanation of possible reasons.

There is a clear correlation between the magnitude of the custom feature loss weights and the angular error ($R^2 = 0.5326$). In particular, high values for λ_{eg} yield lower errors. Concretely, a $\lambda_{eg} = 30$ improved the test error by 53% compared to the baseline with $\lambda_{id} = 2$. However, increasing the feature weights too much breaks the GAN training. For example, when training the model with $\lambda_{eg} = 50$, we observed that the loss of discriminator D_R converged to 0, meaning



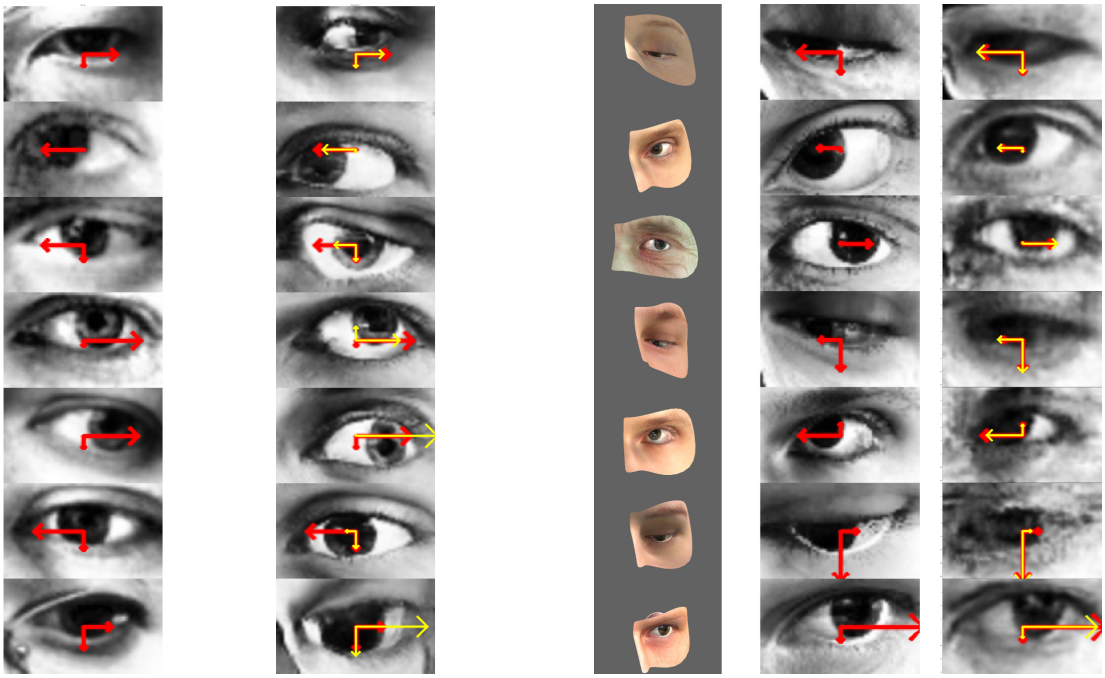
(a) (b)

Figure 3.5.: *Randomly selected images and their translations in both directions for $\lambda_{id} = 2$ and $\lambda_{eg} = \lambda_{lm} = 0$. The arrows indicate the eye gaze direction in pitch and yaw. The red arrow represents the true eye gaze direction and the yellow arrow shows the predicted gaze direction.* (a) *We compare the original image from MPIIGaze (left column) to the produced image (right column).* (b) *The left column contains the original input image from UnityEyes. The middle column shows the pre-processed image and the right column contains the produced version.*

that most of the produced real images were not realistic enough to fool the discriminator.

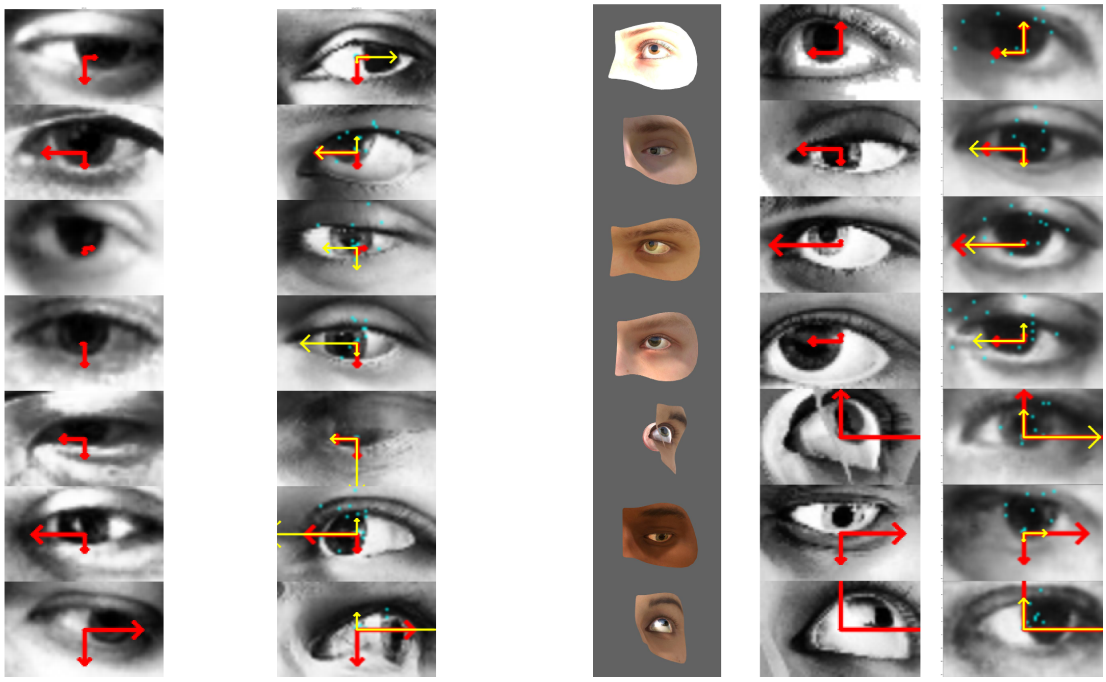
3.4.2. Qualitative Evaluation

For each GAN version, we used the trained generators G and F to translate images and create the produced version. Concretely, G translated images from S to R' and F created the produced versions of images from R in S'. Fig. 3.8 and 3.9 show randomly chosen images and their translations for the GAN models that performed best in the quantitative evaluation. For most GAN versions, we could not differentiate the generated (fake) images from the original images. Fig. **??** showcase randomly selected images and their translations. For the GANs applying an eye gaze feature loss, we drew the original and predicted eye gaze. For the GANs applying a landmarks consistency loss, we additionally display the landmark locations.



(a) (b)

Figure 3.6.: *Randomly selected images and their translations in both directions for* $\lambda_{eg} = 30$ *and* $\lambda_{id} = \lambda_{lm} = 0$. *The arrows indicate the eye gaze direction in pitch and yaw. The red arrow represents the true eye gaze direction and the yellow arrow shows the predicted gaze direction.* (a) *We compare the original image from MPIIGaze (left column) to the produced image (right column).* (b) *The left column contains the original input image from UnityEyes. The middle column shows the pre-processed image and the right column contains the produced version.*



(a) (b)

Figure 3.7.: *Randomly selected images and their translations in both directions for $\lambda_{lm} = 15$ and $\lambda_{id} = \lambda_{eg} = 0$. The arrows indicate the eye gaze direction in pitch and yaw. The red arrow represents the true eye gaze direction and the yellow arrow shows the predicted gaze direction. The blue dots represents the regional landmark coordinates as identified by the landmarks detector. We can clearly see that the landmark detector does not work very well on these images.* (a) *We compare the original image from MPIIGaze (left column) to the produced image (right column).* (b) *The left column contains the original input image from UnityEyes. The middle column shows the pre-processed image and the right column contains the produced version.*

λ_{id}	λ_{eg}	λ_{lm}	*Test Score R2S*
2	0	0	22.9628
5	0	0	GAN failed
10	0	0	GAN failed
0	1	0	18.5872
0	5	0	13.1947
0	8	0	12.0152
0	15	0	11.1988
0	30	0	10.6860
0	50	0	GAN failed (D_R converged to 0)
1	20	0	11.0366 (RHP)
0	30	0	10.4057 (RHP)
1	1	0	18.2186
5	10	0	failed (transl.)
5	15	0	failed (transl.)
1	20	0	11.0366 (RHP)
0	0	1	28.4019
0	0	5	27.5622
0	0	8	27.3049
0	0	15	26.056
0	0	20	GAN failed
0	0	25	GAN failed
0	0	30	GAN failed
0	0	50	GAN failed
0	0	100	GAN failed
5	0	10	GAN failed
10	0	15	GAN failed

Table 3.1.: *Mean angular error for eye gaze estimation in degrees. The model is trained on augmented synthetic images from UnityEyes. For testing, we translate real images from the MPIIGaze dataset to the synthetic domain and test on these translated images. The entries where we used a synthetic dataset with a restricted head pose are marked with* RHP. 'GAN failed' *means that when training the GAN, one of the discriminators converged to* 0. 'Transl.' *indicates that the GAN did not produce images that were visually different to the input images and hence, the translation had failed.*

λ_{id}	λ_{eg}	λ_{lm}	*Test Score R2S*	*Test Score S2R*	*Test Score Within*
2	0	0	22.9628	13.7187	7.1
0	30	0	10.6860	10.6522	6.6625
0	0	15	26.056	23.2255	7.4862

Table 3.2.: *Quantitative evaluation for all three evaluation types (two within and one between). The scores indicate the mean angular error for the given setting. In R2S, we trained on synthetic images and evaluated on produced synthetic images. In the S2R setting, we trained on produced real images and tested on real images from MPIIGaze. The baseline score for a model trained on synthetic data (not applying the FP-GAN) was* 9.8988 *for R2S and* 5.6658 *for within. Hence, the image refinement did not benefit the eye gaze estimation.*

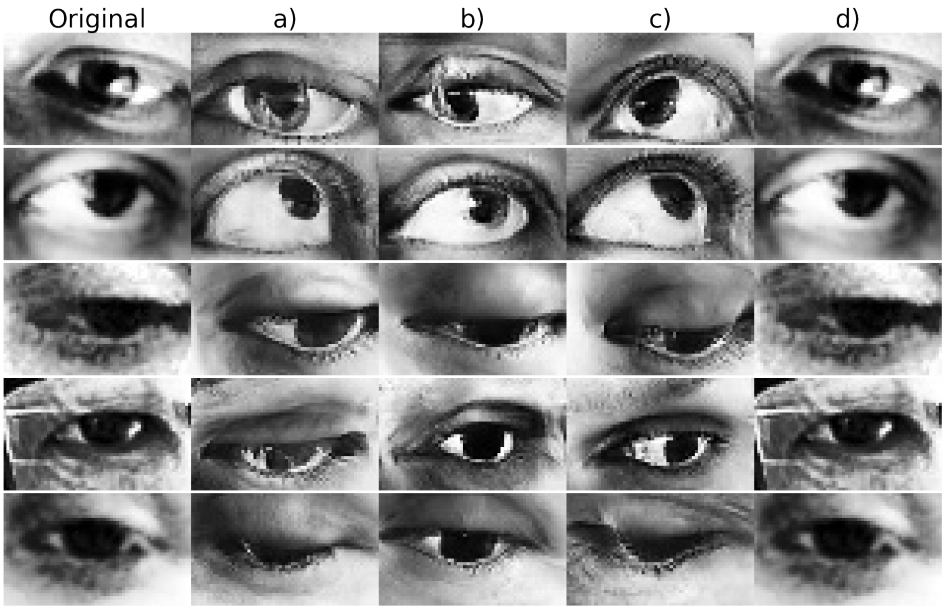


Figure 3.8.: *Qualitative comparison of the translations from the real domain* R *to the produced synthetic domain* S'*. The leftmost column corresponds to real images from the MPIIGaze dataset. Columns a) to d) correspond to the following settings: a)* $\lambda_{id} = 2, \lambda_{eg} = \lambda_{lm} = 0$ *b)* $\lambda_{eg} = 30, \lambda_{id} = \lambda_{lm} = 0$ *c)* $\lambda_{lm} = 15, \lambda_{id} = \lambda_{eg} = 0$ *d)* $\lambda_{id} = 5, \lambda_{eg} = 15, \lambda_{lm} = 0$ *(failed). We observe that in column d), the weights of the feature losses were too high and therefore, the generator did not produce images that were optically different to the input images.*

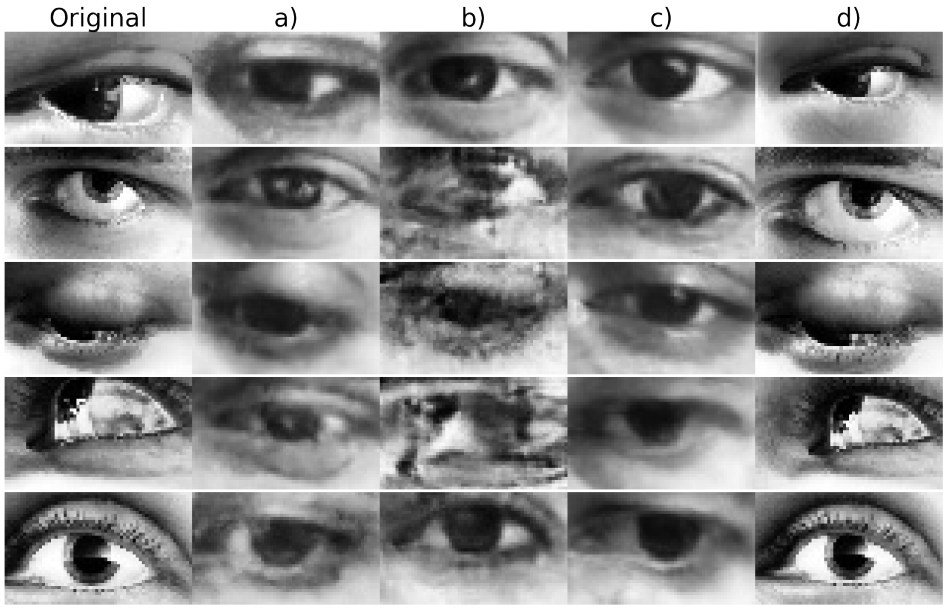


Figure 3.9.: *Qualitative comparison of the translations from the synthetic domain S to the produced real domain R'. The leftmost column corresponds to real images generated by UnityEyes. Columns a) to d) correspond to the following settings: a) $\lambda_{id} = 2, \lambda_{eg} = \lambda_{lm} = 0$ b) $\lambda_{eg} = 30, \lambda_{id} = \lambda_{lm} = 0$ c) $\lambda_{lm} = 15, \lambda_{id} = \lambda_{eg} = 0$ d) $\lambda_{id} = 5, \lambda_{eg} = 15, \lambda_{lm} = 0$ (failed). Again we observe that in column d), the generator did not produce images that were correctly translated to the new domain.*

3.5. Discussion

The baseline eye gaze model trained on heavily augmented synthetic images achieved a similar performance as [Wood et al. 2016]. It performed better on the MPIIGaze dataset than any of the models trained on produced real images. Similarly, the baseline model had a lower error when testing directly on MPIIGaze images compared to testing on refined MPIIGaze images. The refinement of the images by the FP-GAN seemed to have a detrimental effect on eye gaze accuracy.

This seems surprising because firstly, the produced images looked very realistic and secondly, this result does not align with the scores obtained by similar approaches [Shrivastava et al. 2016, Lee et al. 2018]. Possible reasons for that could be technical, e.g. a difference in batch size for training the GAN ([Lee et al. 2018] used a batch size of 64 and we used 8 due to limited memory availability) or constraints on the synthetic input (e.g. Lee et al. generated a synthetic dataset with a restricted range of head poses). It is also possible that the authors of the other paper performed some (to us unknown) pre-processing or fine-tuned the eye gaze models in order to achieve a low error rate.

Another difference between our approach and Lee et al.'s is the dimensionality of the output variable. [Lee et al. 2018] output a three-dimensional vector, whereas our projects estimated eye gaze direction in two dimensions (pitch / yaw).

As an experiment, we trained eye gaze predictors on both normalised and un-normalised target variable (angle in radians). The model did not perform better in any particular case.

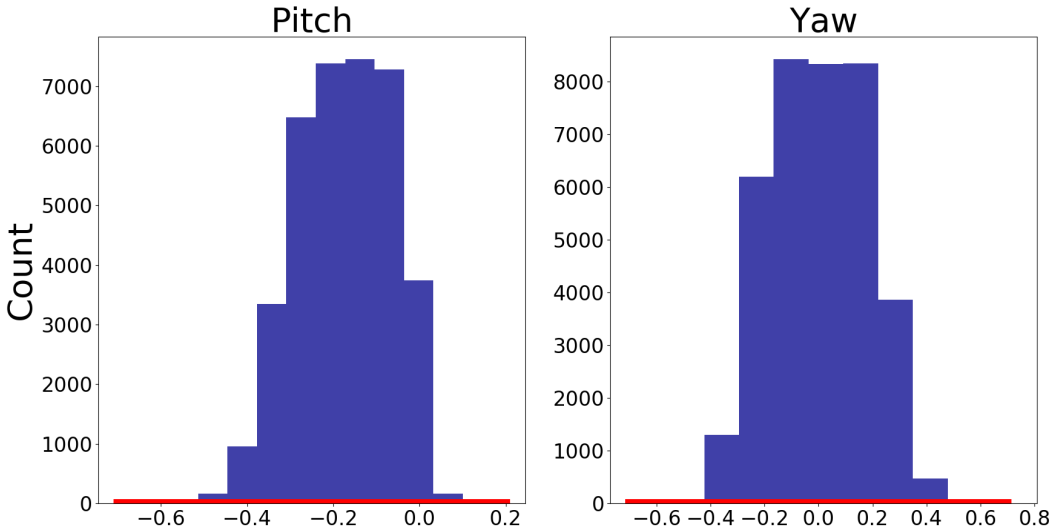


Figure 3.10.: *This histogram shows the distribution of pitch and yaw for the MPIIFaceGaze dataset [Zhang et al. 2017a]. The red bar on the bottom indicates the region that we selected when restricting eye gaze in the UnityEyes dataset.*

We also experimented with a variety of settings in order to find well-performing combinations. For example, we trained GAN and eye gaze models with different input image sizes (120×72 and 60×36). We started working with RGB images and later switched to gray-scale images in order to speed up the experiments. For the RGB variants, we applied the identity transform loss $\mathbb{L}_{id}$ first on RGB and later on gray-scale images, but we did not see a significant difference in model performance. The smaller gray-scale images allowed training the models faster at the cost of slightly inferior results in the quantitative evaluation.

When analysing the outputs of the eye gaze predictors, we found a divergence in distribution between the UnityEyes and MPIIGaze datasets. The MPIIGaze dataset follows an unbalanced distribution with mostly frontal eye gaze directions, whereas the eye gaze directions in the UnityEyes dataset are uniformly distributed. In order to investigate the influence of this fact on the accuracy of our model, we created two datasets for UnityEyes. One dataset was restricted to a frontal head pose, the other dataset allowed the head pose to vary freely. The model trained on the dataset with the restricted head pose is marked as *RHP* in the results tables. The quantitative evaluation did not yield significantly different results.

We also experimented with filtering the eye gaze range when training the GAN, $[-0.7, 0.2]$ for pitch and $[-0.7, 0.7]$ for yaw. This excluded eye gaze directions that did not appear frequently in MPIIGaze. Fig. 3.10 shows the distribution of eye gaze for the MPIIGaze dataset.

We found the models trained on restricted variants to perform slightly better than the ones trained on the other one.

Another aspect we experimented with for the eye gaze estimation models was image augmentation. We applied light and heavy augmentation to the synthetic data. Too light augmentation lead to overfitting the training data. We found that training a baseline model on heavily aug-

mented data to yield very good results when directly tested on real data.

In addition to the variations of the GANs mentioned above, we trained eye gaze estimation models using different learning rates between 1×10^{-4} and 4×10^{-4} and batch sizes between 32 and 256. Furthermore, we built models with ReLU, leaky ReLU and combinations of both activation functions. For all these combinations, we only observed small differences in the model performance. After finding optimal combinations for the UnityEyes dataset, we did not fine-tune further to the translated dataset in order to have a fair comparison between different GAN models. Further improvement by fine-tuning hyper-parameters would certainly be possible.

4

Conclusion and Outlook

The main goal of this work was to investigate the effect of feature preservation for unpaired image translation. Specifically, we re-implemented the architecture proposed by [Lee et al. 2018] with the extension of a feature preserving cost function in tensorflow. We went a step further by preserving eye gaze directions, as well as regional landmarks when training the image translation networks. As a qualitative result, the generated images in the fake domains looked very similar to images in the original domains. Quantitatively, we showed a clear improvement in eye gaze prediction accuracy when putting higher weights on the feature-preservation.

In our re-implementation, we followed [Lee et al. 2018] as closely as possible. Some hyper-parameters (e.g. number of training epochs) were not reported so we chose them in a way such that we had qualitatively satisfying output images. There were some hyper-parameters we could not copy because we did not have the resources (e.g. we did not have enough memory to have the same batch size).

In the quantitative evaluation, we expected our implementation to yield similar quantitative results as [Lee et al. 2018]. However, our models did not yield the same accuracy. It turned out that in our case, the refinement of the images did not help improve the eye gaze estimation. The reasons for this might be technical, for example due to the difference in batch size or lie in the image pre-processing. What we could show, however, was a strong correlation of the feature-loss weights with model performance. Our model trained with $\lambda_{eg} = 30$ improved the angular error by 53% compared to the baseline with $\lambda_{id} = 2$. Lastly, as we were interested in the relative change given different feature loss weights, we did not fine-tune the eye gaze model for each setting. Tweaking hyper-parameters would certainly have improved the absolute model performance and might have yielded an eye gaze predictor that would work well in a real-world system.

Qualitatively, our experiments showed promising results. The generated images looked very

much like images from the true domains. We could not see a big visual difference. We also showed that increasing the weights on the feature loss terms reduced the angular errors for eye gaze estimation for all types of feature loss (identity transform, eye gaze and regional landmark loss). For a more thorough qualitative evaluation, we suggest a user study or using the inception score [Salimans et al. 2016].

Future work could explore the influence of different architectures for the feature loss. In our case, we used a relatively simple, and therefore imperfect Convolutional Neural Network for the eye gaze loss. Having more precise eye gaze estimates could yield lower errors. Similarly, a landmarks detector that generalises well to real data, could yield more accurate results.

A

Appendix

A.1. Implementation on GitHub

For the SimGAN [Shrivastava et al. 2016], several implementations are available online on GitHub. However, we have not found any repository that includes the eye gaze estimator used for the quantitative evaluation.

Our implementation is publicly available on GitHub and it includes the full code for all networks (FP-GAN, feature estimation networks and eye gaze estimator network). We also provide a ready-to-use dataset.
`https://github.com/mbbuehler/FP-GAN`.

A. Appendix

Bibliography

ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G. S., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., GOODFELLOW, I., HARP, A., IRVING, G., ISARD, M., JIA, Y., JOZEFOWICZ, R., KAISER, L., KUDLUR, M., LEVENBERG, J., MANÉ, D., MONGA, R., MOORE, S., MURRAY, D., OLAH, C., SCHUSTER, M., SHLENS, J., STEINER, B., SUTSKEVER, I., TALWAR, K., TUCKER, P., VANHOUCKE, V., VASUDEVAN, V., VIÉGAS, F., VINYALS, O., WARDEN, P., WATTENBERG, M., WICKE, M., YU, Y., AND ZHENG, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

BOLMONT, M., CACIOPPO, J. T., AND CACIOPPO, S. 2014. Love is in the gaze: An eye-tracking study of love and sexual desire. *Psychological Science 25*, 9, 1748–1756.

FABER, M., BIXLER, R., AND K. DÂĂŹMELLO, S. 2017. An automated behavioral measure of mind wandering during computerized reading. *Behavior Research Methods 50* (02).

GATYS, L. A., ECKER, A. S., AND BETHGE, M. 2016. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423.

GLOROT, X., AND BENGIO, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track 9* (01), 249–256.

GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., AND BENGIO, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2672–2680.

HANSEN, D. W., AND JI, Q. 2010. In the eye of the beholder: A survey of models for eyes and gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 3 (March), 478–500.

HUANG, Q., VEERARAGHAVAN, A., AND SABHARWAL, A. 2015. Tabletgaze: A dataset and baseline algorithms for unconstrained appearance-based gaze estimation in mobile tablets.

HUTTON, J. T., NAGEL, J. A., AND LOEWENSON, R. B. 1984. Eye tracking dysfunction in alzheimer-type dementia. *Neurology 34*, 1, 99–99.

ISOLA, P., ZHU, J.-Y., ZHOU, T., AND EFROS, A. A. 2016. Image-to-image translation with conditional adversarial networks. *arxiv*.

JOHNSON, J., ALAHI, A., AND LI, F. 2016. Perceptual losses for real-time style transfer and super-resolution. *CoRR abs/1603.08155*.

KINGMA, D. P., AND BA, J. 2014. Adam: A method for stochastic optimization. *CoRR abs/1412.6980*.

KINGMA, D. P., AND WELLING, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

KRAFKA, K., KHOSLA, A., KELLNHOFER, P., KANNAN, H., BHANDARKAR, S., MATUSIK, W., AND TORRALBA, A. 2016. Eye tracking for everyone. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

KUECHENMEISTER, C. A., LINTON, P. H., MUELLER, T. V., AND WHITE, H. B. 1977. Eye tracking in relation to age, sex, and illness. *Archives of General Psychiatry 34*, 5, 578–579.

KÜHBORTH, K. 2017. *Einfluss genetischer Polymorphismen des CNTNAP2 Gens auf Schizophrenie und kognitive Phänotypen*. PhD thesis, lmu.

KURAUCHI, A., FENG, W., JOSHI, A., MORIMOTO, C., AND BETKE, M. 2016. Eyeswipe: Dwell-free text entry using gaze paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ACM, 1952–1956.

LAGUN, D., HSIEH, C.-H., WEBSTER, D., AND NAVALPAKKAM, V. 2014. Towards better measurement of attention and satisfaction in mobile search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, ACM, New York, NY, USA, SIGIR '14, 113–122.

LEE, K., KIM, H., AND SUH, C. 2018. Simulated+unsupervised learning with adaptive data generation and bidirectional mappings. In *International Conference on Learning Representations*.

LI, Y., XU, P., LAGUN, D., AND NAVALPAKKAM, V. 2017. Towards measuring and inferring user interest from gaze. In *Proceedings of the 26th International Conference on World Wide Web Companion*, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, WWW '17 Companion, 525–533.

LU, F., SUGANO, Y., OKABE, T., AND SATO, Y. 2011. Inferring human gaze from appearance via adaptive linear regression. In *2011 International Conference on Computer Vision*, 153–160.

LU, F., SUGANO, Y., OKABE, T., AND SATO, Y. 2014. Adaptive linear regression for appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 10 (Oct), 2033–2046.

MAO, X., LI, Q., XIE, H., LAU, R. Y. K., AND WANG, Z. 2016. Multi-class generative adversarial networks with the L2 loss function. *CoRR abs/1611.04076*.

MOTT, M. E., WILLIAMS, S., WOBBROCK, J. O., AND MORRIS, M. R. 2017. Improving dwell-based gaze typing with dynamic, cascading dwell times. ACM.

NEWELL, A., YANG, K., AND DENG, J. 2016. Stacked hourglass networks for human pose estimation. *CoRR abs/1603.06937*.

PARK, S., ZHANG, X., BULLING, A., AND HILLIGES, O. 2018. Learning to find eye region landmarks for remote gaze estimation in unconstrained settings. *CoRR abs/1805.04771*.

RADFORD, A., METZ, L., AND CHINTALA, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR abs/1511.06434*.

REZENDE, D. J., MOHAMED, S., AND WIERSTRA, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

SALIMANS, T., GOODFELLOW, I. J., ZAREMBA, W., CHEUNG, V., RADFORD, A., AND CHEN, X. 2016. Improved techniques for training gans. *CoRR abs/1606.03498*.

SATTAR, H., MULLER, S., FRITZ, M., AND BULLING, A. 2015. Prediction of search targets from fixations in open-world settings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 981–990.

SELA, M., XU, P., HE, J., NAVALPAKKAM, V., AND LAGUN, D. 2017. Gazegan - unpaired adversarial image generation for gaze estimation. *CoRR abs/1711.09767*.

SHRIVASTAVA, A., PFISTER, T., TUZEL, O., SUSSKIND, J., WANG, W., AND WEBB, R. 2016. Learning from simulated and unsupervised images through adversarial training. *CoRR abs/1612.07828*.

ULYANOV, D., VEDALDI, A., AND LEMPITSKY, V. S. 2016. Instance normalization: The missing ingredient for fast stylization. *CoRR abs/1607.08022*.

WOOD, E., BALTRUŠAITIS, T., MORENCY, L.-P., ROBINSON, P., AND BULLING, A. 2016. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, 131–138.

ZHANG, Y., MÃIJLLER, J., KI CHONG, M., BULLING, A., AND GELLERSEN, H. 2014. Gazehorizon: Enabling passers-by to interact with public displays by gaze. *UbiComp 2014 - Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (09), 559–563.

ZHANG, X., SUGANO, Y., FRITZ, M., AND BULLING, A. 2015. Appearance-based gaze estimation in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4511–4520.

ZHANG, X., SUGANO, Y., FRITZ, M., AND BULLING, A. 2017. It's written all over your face: Full-face appearance-based gaze estimation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, IEEE, 2299–2308.

ZHANG, X., SUGANO, Y., FRITZ, M., AND BULLING, A. 2017. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *CoRR abs/1711.09017*.

ZHOU, X., TANG, F., GUAN, Q., AND HUA, M. 2017. A survey of 3d eye model based gaze tracking. *Jisuanji Fuzhu Sheji Yu Tuxingxue Xuebao/Journal of Computer-Aided Design and Computer Graphics 29* (09), 1579–1589.

ZHU, J., PARK, T., ISOLA, P., AND EFROS, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR abs/1703.10593*.