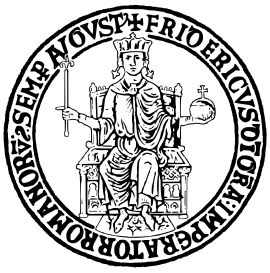
Università degli Studi di Napoli Federico II



Scuola Politecnica e delle Scienze di Base

Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione

Corso di Laurea Magistrale in Informatica

CNN-based transcoding from visible to near-infrared images for improved iris segmentation and recognition

Relatore
Prof. Daniel Riccio

Correlatore
Prof. Antonio Origlia

Candidato
Salvatore Davide Amodio
N97000369

Anno Accademico 2022–2023

A nonno Giovanni

Abstract

This thesis introduces an innovative deep-learning approach to enhance image quality within the iris recognition pipeline operating in the visible (VIS) spectrum. The study leverages the inherent correlation between visible and near-infrared (NIR) spectra, proposing two transcoding models with the purpose of (1) mitigating noise factors in iris images, such as reflection and glare, to improve iris segmentation, and (2) refining normalized (segmented) iris images for finely detailed texture representation, to result in more discriminating features. In its concluding phase, the study introduces a novel feature extraction method that generates high-quality features by exploiting the VIS-NIR spectra relationship, similar to the transcoding techniques outlined. The convolutional neural network (CNN) architectures draw inspiration from U-Net and Pix2pix, well-known in Image-to-Image (I2I) translation tasks. This work attempts to contribute to the research of effective methods to improve the critical steps involved in iris recognition systems by exploiting the benefits of the NIR domain without directly using near-infrared cameras.

Sommario

La seguente tesi propone un innovativo approccio di deep-learning per migliorare la qualità dell'immagine all'interno della pipeline dei sistemi di riconoscimento dell'iride che operano nello spettro visibile (VIS). Lo studio approfondisce la correlazione intrinseca tra gli spettri del visibile e del vicino infrarosso (NIR), proponendo due modelli di transcodifica con lo scopo di (1) mitigare i fattori di disturbo nelle immagini dell'iride, come il riflesso ed il bagliore, per migliorare la segmentazione dell'iride, e (2) perfezionare le immagini normalizzate (segmentate) dell'iride per una rappresentazione più dettagliata della texture, per ottenere feature più discriminanti. Nella fase conclusiva, lo studio introduce un nuovo metodo di feature extraction che genera feature di alta qualità sfruttando la relazione degli spettri VIS-NIR, analogamente alle tecniche di transcodifica descritte. Le architetture delle reti neurali convoluzionali (CNN) traggono ispirazione da U-Net e Pix2pix, ben note nei task di traduzione Image-to-Image (I2I). Questo lavoro cerca di contribuire alla ricerca di metodi efficaci per migliorare le fasi critiche coinvolte nei sistemi di riconoscimento dell'iride sfruttando i vantaggi del dominio NIR senza ricorrere necessariamente a sistemi di acquisizione ad infrarossi.

Contents

Introduction

Biometric identification has become a popular method for replacing traditional identification methods such as passwords, access cards, or keys. However, some biometric traits such as the face, palm print, finger veins, voice, or fingerprint have limitations that affect their usage. For instance, some biometric traits may require a sample, manual setting, or physical contact, which can impact the comfort and user-friendliness of recognition systems. In [1], Jain et al. identified seven factors that each biometric trait should have to be suitable for biometric applications: Universality, Uniqueness, Permanence, Measurability, Performance, Acceptability, and Circumvention. Among all biological traits, the iris pattern is the most accurate and reliable for five main reasons: (1) it has the highest level of uniqueness, even among twins; (2) it is stable over a lifetime; (3) it is informative due to its complex and rich texture; (4) it is difficult to counterfeit; and (5) it is contactless, which is highly appreciated by users [2]. Numerous studies have shown that the near-infrared (NIR) spectrum is more effective than the visible (VIS) spectrum for iris texture extraction, making it the preferred choice for most iris recognition systems. The NIR spectrum offers two significant advantages: (1) the texture of the iris is not distorted during NIR acquisition because the pupil is not stimulated by NIR illumination, and (2) NIR acquisitions are better suited for capturing the texture and morphology of dark-colored irises [3]. However, NIR-based recognition systems have limitations, particularly in close-range acquisition scenarios. As a result, researchers have continued to explore methods in the VIS spectrum, which has proven to be highly efficient at increased distances and is more cost-effective due to the availability of low-cost visible light cameras [4]. The VIS spectrum is necessary in specific contexts, such as video surveillance environments, especially when cameras or webcams lack NIR sensors. Improved performance in iris recognition with the visible spectrum is highly beneficial, especially for subjects with dark-colored irises, which pose a significant challenge.

This study investigates the feasibility of translating VIS images into the NIR spectrum to explore the relationship between the two spectra by constructing transcoding models to apply an Image-to-Image (I2I) translation task; I2I is a computer vision technique that transfers images from a source domain to a target domain while preserving content representations [5]. Various applications have applied I2I, including image synthesis,

segmentation, style transfer, restoration, and pose estimation. The translation provides synthesized images in the NIR spectrum starting from its original VIS representation. It aids in improving iris image quality, leading to more accurate results in the workflow of iris recognition systems.

The first chapter explores the latest techniques for iris recognition systems, highlighting their strengths and weaknesses and comparing the traditional methods to newer machine learning approaches. The second chapter delves into Image-to-Image translation applications and various techniques for tackling this challenge.

The third chapter shifts to the challenges associated with iris segmentation in visible light and the benefits of converting images to the NIR spectrum. Thus, the chapter proposes architectures to design a transcoding model, improving image quality and segmentation accuracy. In contrast, the fourth chapter emphasizes the more pronounced features of the iris in the NIR spectrum. It suggests a transcoding method to enhance the texture quality of iris images captured in visible light to improve the iris pattern's clarity and distinctiveness. The last section proposes a novel feature extraction method inspired by a Daugman-based approach.

The fifth chapter provides an overview of the development environment, proposed models, training details, dataset used, metrics for evaluating patterns, and results obtained from testing. The performance of an iris recognition system is ultimately compared using the original images versus the highest quality images obtained through the two best-performing models in the proposed tasks.

The last chapter presents the conclusion of this work and possible future developments.

–1–
Iris Recognition Systems

In the Introduction, iris recognition is presented as the most reliable and secure technique for personal authentication. However, the iris is one of the most complex biological traits to locate and capture. The iris is a protected internal organ of the eye between the cornea and the lens. It is made up of muscular tissue, including a sphincter muscle (to contract the pupil) and a collection of dilator muscles (to dilate the pupil). Typically, a human iris is small in size (around 11mm in diameter) and contains many minutiae such as stripes, furrows, collarettes, freckles, stroma, coronas, and crypts [6].

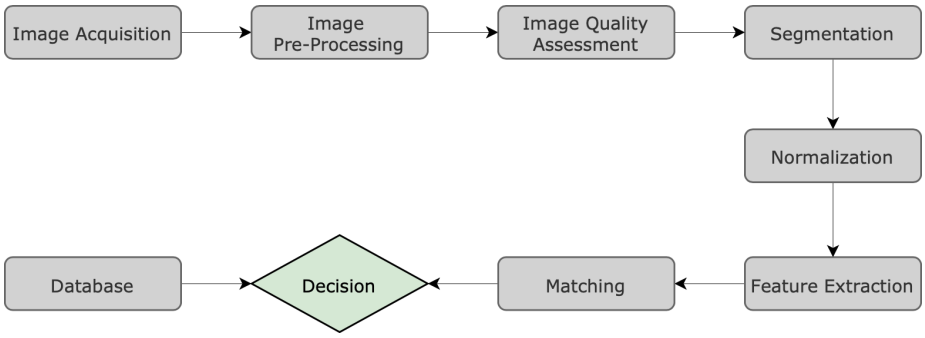


Figure 1.1: Architecture of an iris recognition system.

Flow and Safir patented the first concept of an automated iris recognition system [7]. A few years later, Daugman presented his approach, which is today considered conventional, as most current solutions have relied on Daugman's method of iris recognition [8]. Figure 1.1 illustrates the classical architecture of an iris recognition system, which consists of six modules: iris image acquisition, iris image localization, iris image normalization, feature extraction, matching, and decision. The following sections describe each step in detail, presenting old and more up-to-date strategies that have been implemented to overcome the weaknesses and shortcomings shown over time.

1.1 Image acquisition

Most acquisition systems have used the near-infrared (NIR) band to capture iris images. The iris can be very pigmented (especially in dark-colored irises), which interferes with the capture of iris texture; a wavelength between 700 and 900 nm helps this process. Several studies have shown that acquiring images in the NIR spectrum has led to a higher recognition accuracy than in the visible light (VIS) spectrum (especially in a controlled environment), as it is tough to distinguish the richness of the iris texture in VIS images. Nevertheless, the VIS band has some potential: investigations have shown that iris recognition appears efficient at increased distances, even without complicated sensors [8]. This has led to in-depth studies on the performance of iris recognition systems with VIS images and several competitions, such as the Noisy Iris Challenge Evaluation (NICE) [9] and the Mobile Iris Challenge Evaluation (MICE) [10], have provided datasets in VIS spectrum.

1.2 Image pre-processing

In the iris recognition system, it may be crucial to have an image pre-processing phase because image quality strongly impacts iris recognition. Occlusion may be the main noise factor involving low quality caused by eyelids or eyelashes. However, other factors can compromise a good capture: light or illumination, specular reflections, noise due to eyeglasses (blurring), scratches, distortion, contact lenses, and off-axis irises [11].

In the literature, four main approaches to remove noise have been proposed as pre-processing methods: (1) Hough transform method: to remove noise related to eyelashes, eyelids and to handle reflections due to eyeglasses or contact lenses, researchers have first tried with Hough transform methods and then algorithms that better localize the pupil region; (2) histogram and filtering method: it has been mainly applied to enhance image quality by redistributing pixel intensities in those images with low contrast and low illumination. It has flavored the pupil localization, increasing the iris segmentation rate; (3) morphological operations: those operations have been massively used to remove noise caused by the same factors described in the first method; however, those operations have dropped their performance in non-ideal environments; (4) fusion method: the method proposed by Li and Ma localizes non-circular iris boundaries acquired in an unconstrained imaging environment through the RANSAC algorithm. Other approaches include using the line intensity profile (LIP), applying SVM, or using morphological operations to identify, classify, or remove reflections. Other approaches include normalization of illumination using the SSR technique, histogram equalization, and bilateral filtering to reduce noise and improve the accuracy of iris segmentation [11].

1.3 Image quality assessment

Even though image quality assessment is not considered fundamental, this step is relevant in any iris recognition system because image quality significantly affects performance. Poor-quality images enlarge intra-class variability and reduce inter-class variability, increasing FRR and FAR. An image can be affected by some main factors: defocus, motion blur, or occlusion. Some algorithms, such as that proposed by Wei et al., select the clearest image among an image sequence by considering the three factors mentioned above as main noise; the more defocus, motion blur, and occlusion are contained within the image, the more it will be considered to be of poor quality [12]. Nevertheless, due to the importance of the topic, more complex algorithms explore more factors: in [13], leaving aside defocus blur, motion blur, and occlusion, the algorithm the author proposed bears in mind lighting, specular reflection, off-angle, and pixel-counts on the performance of traditional iris recognition system. For them, defocus blur, motion blur, and off-angle most affect the recognition performance.

1.4 Segmentation

At this stage, the goal is to isolate the iris region from the eye image, detaching it from the pupil region and all the artifacts around the iris, such as eyelids, sclera, and skin. The minimum error is essential to achieve a good feature extraction stage.

1.4.1 Traditional techniques

Daugman and Wildes's approaches have long been guidelines for the segmentation task. The former used an integro-differential operator, while the latter suggested a method involving Hough transform (HT) and edge detection. The HT finds circles to be the best candidates to represent the pupillary and limbic boundaries, and this method has become very popular despite its limitations in less-constrained environments when the reflection, the occlusion, and the shadow are quite present within the image. Consequently, some works have focused on these factors to better segment the iris region.

Pupillary and limbic boundaries

In the research, various methods have been proposed to improve Wildes et al.'s implementation [14] based on edge detection and Hough transform. Liu et al. [15] simplified Huang et al.'s methods [16] by finding the iris on a rescaled image and projecting this information of the real image the Canny edge detection, making it faster; they assumed pupillary and

limbic boundaries as concentric circles. Liu et al. [17] introduced the 'ND_IRIS' segmentation algorithm, removing edge points around specular highlights and implementing an improved Hough transform; they achieved a location rate of 97.1%. Some approaches have focused on coarse pupil localization, such as the histogram-based method of Lili and Mei [18] or the 'coarse to fine' strategy of Feng et al. [19]. Other works have achieved high performance in tests on the CASIA 1 dataset, assuming the pupil to be a completely dark and uniform region. A recent trend has been to process off-angle iris images. The aim is to rotate the images by projecting them onto a frontal view, knowing the angle of rotation [20].

Occlusion by eyelids, eyelashes, and specularities

Kong and Zhang [21] handled iris segmentation by considering two kinds of occlusion: *separable* and *mixed* eyelashes. The former allows to detach eyelashes from iris texture, while the latter causes larger occlusions. They modified Boles' method [22] and reduced the Equal Error Rate (EER) on the dataset that Boles used to conduct his experiment by up to 3%. Huang et al. also tackled occlusion issues caused by eyelids, eyelashes, and specular highlights. They used phase congruency-based edge information to identify occlusion regions. Their experiments with the CASIA dataset improved the Receiver Operating Characteristic (ROC) curve in recognition evaluation. Several open topics are still active in segmentation steps, such as: (1) not approximating pupillary and limbic boundaries as circles; (2) robust segmentation in subjects with contact lenses and glasses [20].

1.4.2 Deep learning techniques

In recent years, three deep learning models have been particularly involved in segmentation tasks: U-Net architecture-based models, VGG and R-CNN architecture-based models, and other CNN architecture-based models [11].

U-Net architecture-based

In [23], Lian et al. proposed the Attention U-Net (ATT-UNet), which is based on the primary U-Net with an attention layer helping to classify better pixels that belong to the iris region and the pixels that do not. However, limitations include occlusions such as hair, eyelashes, eyeglasses, reflections, low illumination, and blurring. More information needs to be extracted from the iris image through more informative features; therefore, Zhang et al. [24] proposed four schemes combining dilated convolution and U-Net to enhance segmentation performance. Other techniques have been tabled to handle unconstrained conditions, such as Dense U-Net by Wu and Zhao [25]. In addition, Wang et al. [26]

introduced IrisParseNet, to address challenges such as different-sized irises, occlusions, illumination, and specular reflections.

VGG and R-CNN architecture-based

IPSegNet1 and IPSegNet2 of Patil et al. [27], inspired by SSD and R-CNN, apply pupil and iris segmentation together to obtain a circular region, but they have limitations in the presence of non-ideal iris images, including off-angle views and eyelash occlusions. Rot et al. [28] adopted the SegNet architecture, an FCN for a semantic segmentation function, to divide eye images into different regions, addressing multiple challenges but not all. Korobkin et al. [29] combined FCN and SegNet architectures to enhance segmentation in various conditions but faced occlusion and off-angle issues. Arsalan et al. [30] introduced IrisDenseNet, combining DenseNet and SegNet techniques, but faced problems of false positives and negatives due to eyelash and noise. Arsalan et al. [31] introduced FRED-Net, an end-to-end semantic segmentation network based on SegNet, to improve segmentation performance. However, his model highly depends on the number of training images (false positives and negatives). These limitations encouraged Li et al. [32] to overcome them, combining Faster R-CNN and Gaussian mixture model (GMM) for pupillary detection and limbus edge localization. Zhao and Kumar [33] proposed UniNet.v2 using Mask R-CNN for iris region detection and segmentation. The net achieves higher accuracy compared to previous approaches.

Generic CNN architecture-based

Liu et al. [34] introduced two networks, MFCNs and HCNNs, for iris segmentation in at-a-distance and on-the-motion environments. MFCN outperforms HCNN but has limitations, as it misclassifies non-iris pixels as iris pixels. To address this problem, He et al. [35] proposed a deep CNN based on DeepLab to extract eye features and segment the iris, pupil, and sclera. The limitations of this approach are several: image contrast, image blurring, noise level, and eye resolution. Thus, Lozej et al. [36] used DeepLabV3 with MobileNet to segment heterogeneous iris images but faced challenges with thin eyelashes. Further research should prioritize two aspects: (1) new network architectures and a different augmentation step; (2) better handling of borderline cases such as images of iris diseases [11].

1.5 Normalization

This phase concerns converting the circular iris region into a rectangular pattern of fixed dimensions to overcome problems related to different dimensions of iris regions. Most papers have cited Daugman's method that maps each point of the iris region to a pair

(r, θ) in a polar coordinate system where r is the radius in $[0, 1]$ and the angle θ is in the range $[0, 2\pi]$. [37]. However, this approach is very sensitive to rotation. Some works have tried to solve this limitation, such as Shamsi and Rasouli with a trapezoidal approach [38], but the rubber-sheet model is still broadly popular.

1.6 Feature extraction

Feature extraction from iris texture is critical to creating biometric templates that efficiently represent individuals and produce successful iris recognition performance. As for iris image segmentation, the advent of machine learning has strongly led researchers to adapt it to this step, with significant results.

1.6.1 Traditional techniques

A first approach, also adopted by Daugman, involves using Gabor filters. However, the Gabor wavelets have a limitation in processing spectral information with maximal spatial information. Several researchers have turned to 1D log-Gabor filters and, later, 2D log-Gabor filters, which encode iris images by better representing high frequencies under-presented by the traditional Gabor filters method. In particular, 2D log-Gabor filters have been massively used to extract global tissue information from the iris regions [11]. Another approach that produces better spatial and spectral localization is the Discrete transform wavelet (DTW), as demonstrated by Kumar et al. [39] and Kekre et al. [40], who based their feature extraction step on DTW. Still, they needed help in real-time applications or unconstrained environments. The scale-invariant feature transform (SIFT) invented by Lowe [41] handles noisy environments but needs to be improved in terms of time and accuracy rate. Barpanda et al. proposed two methods for iris feature extraction: the first method relies on a wavelet derived from the popular biorthogonal Cohen-Daubechies-Feauveau 9/7 filter bank to extract features in three steps [42] while the second method relies on wavelet mel-cepstrum wavelets [43]. As a result, the second method needed to improve its accuracy with decreased feature size compared to the first method.

1.6.2 Deep learning techniques

The main shortcoming of traditional approaches is the significant effort in pre-processing and parameter tuning to obtain discrete results with a given dataset. Thus, changing datasets can compromise guaranteed performance. With a deep learning approach, the wish has been to find a strategy that extracted some general features that could be transferred to different tasks. There was a breakthrough when Krizhevsky et al. [44] proposed a new architecture called AlexNet that achieved higher performance in ImageNet

Large Scale Visual Recognition Competition (ILSVRC) by leveraging the rectified linear unit (ReLu) as the activation function. Over time, numerous architectures have been implemented, mainly in an attempt to beat the competition-winning models, such as ZFNet, GoogLeNet/Inception, or VGGNet. In [45], the authors demonstrated how features learned while training a neural network for image recognition can be applied to different tasks and datasets to produce impressive results. Since then, all of the mentioned architectures have been involved in the iris recognition task. The trained model has often been treated as a feature extraction engine to extract high-quality features from iris images. Many works have driven this trend; researchers have tried to experiment more, exploiting the trained models and mixing them with something new to achieve even better performance in the iris recognition task. For instance, VGG networks have been involved in many works for deep feature extraction with different configurations: initially Simonyan and Zisserman [46] suggested a VGG architecture with 16 layers (VGG16), then other researchers as Carvalho et al. [47] proposed VGG architecture with 19 layers (VGG19) while Minaee et al. [48] applied Principal Component Analysis (PCA) to reduce the dimensionality of features extracted with a VGG network. However, In research, there have likewise been works where architectures have been designed from scratch, such as OcularNet, a convolutional neural network (CNN) model for mobile ocular biometrics that uses patches of eye images [49].

1.7 Matching

Once the feature extraction phase is completed, the new template must be compared with the stored templates to measure similarities (expressed as scores) and determine the generated template's identity. Often, due to the elevated numbers of identities, a full match is not feasible; therefore, the matching step is replaced with the classification step. In [50], an SVM model was developed to classify each biometric template into one of two classes: accept or reject person. The authors chose the kernel that best represented and isolated the data by decision boundary in their task. Instead, Rana et al. [51] applied the k-nearest classifier using Euclidean distance as a distance metric. In this case, the classifier takes as input a biometric template and is classified or recognized according to the classes its k-nearest neighbours belong to; it is assigned to the class most common among its k-nearest neighbours. This approach has been practical in many works, and sometimes, the 1-neighbour classifier is applied, in which the template is assigned to the class of the nearest neighbour. Daouk et al. [52], Chirchi et al. [53], Yiming et al. [54], and so many researchers have been using the Hamming Distance (HD) to match two iris templates, proving that HD is still one of the most efficient metrics among traditional methods. Hanfei and Congfeng [55] published their work proposing a new metric called Hamming Distance Deviation Matching Approach (HDDMA) for iris recognition, claiming

better EER in experimental tests performed on different datasets. However, deep learning techniques have gained much attention, and several attempts have been made to achieve even better performance in classification, including trying to combine different state-of-the-art architectures, such as UniNet, a network unifying FeatNet and MaskNet, proposed by Zhao and Kumar [56]. Other work has focused on using well-known, pre-trained networks such as AlexNet to classify features from the feature extraction stage [57].

–2–
Image-to-Image Translation

Zhu J. et al. defined the Image-to-image translation (I2I) as "a class of vision and graphics problems in which the goal is to learn the mapping between an input image and an output using a training set of aligned image pairs" [58].

The goal is to learn how to map an image x_A, which belongs to a source domain A, into a target domain B, resulting in x_{AB}. The mapping $G_{A \to B}$ must preserve in x_{AB} the intrinsic content of the A domain, contained in x_A, but with the extrinsic style of the B domain. Mathematically, this process is summarised by this formula [5]:

$$x_{AB} \in B : x_{AB} = G_{A \to B}(x_A)$$

The transformation impacts the representation while the concept does not have to change (i.e., edge maps, image colorization, depth map estimation). Nevertheless, I2I is likewise used to remove unwanted features, noise, and artifacts. Thus, the representation remains intact, and the transformation affects the domain. I2I has been considered one of the most challenging problems in computer vision and has also been addressed in many image processing and computer graphics tasks. In recent years, I2I has been brilliantly applied in particular to image synthesis [59], [60], image segmentation [61], image super-resolution [62] and style transfer [58], [63], [64].

2.1 Backbone of I2I

I2I and generative models are closely related. Translating images from a source domain to a target domain involves learning the mapping between the domains to correctly represent the images in the target domain, while the purpose of a generative model, assuming a probability distribution represents the data, is to estimate and approximate it to generate a distribution very similar to the original one [65]. Ng and Jordan declared that "generative classifiers learn a model of the joint probability, $P(X, Y)$, of the inputs X and the label

Y" [66]. In mathematical terms, considering X and Y as the independent and the target variables, respectively, $P(X|Y)$ and $P(Y)$ are estimated by a generative model. Additionally, it is possible to calculate $P(Y|X)$ using Bayes' theorem, which results in a Naïve Bayes classifier; this classifier is considered to be generative [65].

Therefore, looking again at the relation between I2I and generative models, an I2I task can be solved by modeling a generative model to approximate the distribution of the target domain; all 'fake' data generated by the model are all translated images that belong to the distribution of the target domain. In the following section, variational autoencoders (VARs) and generative adversarial networks (GANs) are described in detail, as these are the two significant families that deserve more attention because they offer outstanding performance. The GAN architecture has become the primary strategy for solving whatever I2I tasks in research due to its different approach. The old formulation of the problem involved considering the output space as "unstructured", meaning that all the pixels contained in the output image are conditionally independent. Instead, with GAN formulation, the output space is "structured"; loss decisively affects the joint configuration of the output image [67].

2.2 Variational AutoEncoders

Knigma and Welling proposed the variational autoencoder (VAE), a sophisticated technique to compress and represent data in a lower dimensional space. A VAE is an artificial neural network architecture to describe, probabilistically, an observation in a latent space. Their algorithm is scalable over a large dataset and works in the intractable case (under certain mild conditions [68]). The primary purpose of a VAE is to guide the training process to obtain a latent space with suitable properties to generate new high-quality data. The trick is to regularise the encoding distribution; the term 'variational' comes from the relation between regularisation and variational inference method [69].

The backbone of a VAE is an autoencoder, which, as the name suggests, is designed to learn how to encode data in a different representation. It consists of two neural network units: an encoder and a decoder. The former handles the data encoding process unsupervised to produce a new representation that compresses the data. In machine learning, the process of representing the data in a reduced number of variables (features) is called *dimensionality reduction*, and the space in which the encoder compresses the data is called *encoded space* or *latent space*. The latter handles the decompression phase, the reverse process leading to the data generation from the latent space. This architecture presents a bottleneck layer, which can result in lossy compression. It implies that, during the encoding-decoding transition, the encoder cannot compress the information, and the decoder cannot wholly reconstruct it. Besides being a brilliant architecture for dimensionality reduction, autoencoders provide a generative capability, which is the motivation

why this architecture is proposed in this chapter. After training the model, the latent space should represent the original domain in a lower dimensional space, retaining only the valuable information and properties of the data distribution. Therefore, the general idea is to randomly sample a point in the latent space to obtain new content. The decoder function acts like a generator in GAN architecture. However, autoencoders suffer from a substantial limitation; if the latent space is not well-organized, they cannot generate data, which highly depends on the underlying data distribution. The variational autoencoders are invented to regularize the latent space to provide an accurate and stable generative capability.

2.3 Generative Adversarial Networks

Although VARs generate images very close to the original distribution, they suffer from the problem of blurring, resulting in inaccurate images [65]. A new framework, inspired by a two-player minimax game, is stealing the scene in the world of machine learning; it is called Generative Adversarial Network (GAN). The idea of Goodfellow et al. [70] is so revolutionary that countless works based on this proposal generate numerous variants with many improvements, theoretical extensions and revisions of the initial model [71]. The original idea consists of an adversarial process that simultaneously trains two models: a generative model G and a discriminator model D. The former is the generator that captures the data distribution, generating fake data from stochastic noise. At the same time, the latter estimates the probability that data comes from either the generator or the real data distribution (so it distinguishes between real and fake data. G and D act as two players; thus, the aim is to find a Nash equilibrium defining the value function $V(D, G)$ of the minimax game. Formally, G is a generator network and models a differentiable function that inputs random noise z, sampled from the latent space Z following the distribution $p_z(z)$, and outputs fake data hoping that it belongs to the original data distribution $p_{data}(x)$:

$$G \ : \ Z \rightarrow R^n$$

D is a classifier neural network and models a function that maps from the data distribution to a probability $p \in [0, 1]$. In other words, D is a discriminator that distinguishes between authentic and fake data by labeling each input data with p expressing how real the input is [72]:

$$D \ : \ R^n \rightarrow [0, 1]$$

As anticipated before, D is trained to maximize the probability of correctly labeling both the training data and the generated samples. In contrast, G is trained to minimise the $\log(1 - D(G(z)))$. Hence, the objective optimization problem can be expressed as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.1}$$

GANs contribute massively to image generation tasks but present some limitations, as might be expected. The main drawback concerns the training process, which is very complicated to implement. GAN models usually suffer from (1) non-convergence: the algorithm gets stuck in local minima or has problems with slow convergence; (2) mode collapse: the generator only produces a subset of the real data distribution; (3) vanishing gradients: during backpropagation, the gradients approach zero leaving the weights of the lower layers unchanged [73].

2.3.1 Conditional GANs

The previously described model can be considered an *unconditional GAN*, as an image is generated from a random noise input z. Thus, the generation is uncontrolled; in [74], the authors proposed to add condition information y to address the image generation by creating a *conditional GAN*. The objective function is:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}}[\log D(x|y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y)))] \tag{2.2}$$

Any auxiliary information can be represented by y: text, images, and data labels.

Pix2Pix Among all the proposed conditional GAN architectures, Pix2pix architecture [67] offers a general-purpose solution to image-to-image translations (although, as the authors specified, it remains unclear how to truly generalize for all types of tasks; they demonstrated how Pix2pix model synthesizes images and drastically reduce the problem of blurring. The authors chose to implement Pix2pix model in a way that differs entirely from other works. A U-Net-based architecture implements the generator module, while a 'Patch-GAN' classifier characterizes the discriminator module. The U-net was introduced by Ronneberger et al. [75] for a biomedical image segmentation task.

Any U-net shapes an encoder-decoder architecture [76]; in Ronneberger's work, the encoder aims to extract relevant features from images, and the decoder aims to reconstruct the segmentation masks. In his paper [75], the left-side (encoder) and the right-side (decoder) are called *contracting path* and *expansive path*, respectively. As mentioned, many image-generating jobs suffer from blurry L1 or L2 loss results. However, these losses capture the low frequencies that represent the core of the information, so they are nonetheless used to complete these tasks. Their flaw is solved by including a discriminator that models high frequencies and analyses the image for local patches. Isola et al. named

the discriminator architecture *PatchGAN* that "only penalizes structure at the scale of patches" [67]. Each $N \times N$ patch is labelled as real or fake. A PatchGAN is a convolutional neural network, and its output indicates the likelihood that the image patches are real or artificially created [77]. The discriminator can accurately identify an image from a real distribution or a generator network output by paying attention to local features combined with global structure. The paper describes a conditional GAN as an architecture that leverages the mapping from the observed image x and the random noise vector z to y. Therefore, a conditional GAN loss function can be expressed as:

$$\begin{aligned} \mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \\ \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))] \end{aligned} \tag{2.3}$$

where the generator function is $G : \{x, z\} \rightarrow y$ and an L1 loss function is adopted because it seems to favour less blurring:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||_1] \tag{2.4}$$

the objective function can be expressed as follows:

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{2.5}$$

Through various experiments, Isola et al. proved how the reduction of artifacts was achieved by setting $\lambda = 100$ [67].

2.4 Evaluation Metrics

In I2I tasks, the evaluation phase is concerned with quantitative and qualitative estimation of the translation process, although this is still an open problem. Traditional metrics do not consider the structure that models try to transfer within the image. In addition, the use of different metrics can lead to conflicting results because each metric may focus on different aspects to assess image quality, and the visual quality of a digital image can be subjective. Therefore, several metrics have been proposed to evaluate generated images, commonly split into subjective and objective metrics.

Amazon Mechanical Turk (AMT) This is the only subjective metric presented in this work; the accuracy and the authenticity of generated images are given by human perception. Workers (or 'turkers') are instructed to select or evaluate the best image based on quality and perceptual realism after receiving an image input and translated images [78].

Peak signal-to-noise ratio (PSNR) The PSNR is adopted to measure the intensity variations between the translated and original images; The PSNR is calculated based on the ratio of the maximum possible power of a signal and the power of corrupting noise that affects the quality of its representation. This ratio is often a good indicator in compression tasks where the more the generated images are similar, the higher the PSNR score and the quality of compression.

Structural similarity index (SSIM) The SSIM has been one of the major metrics used in image quality assessment over the years, although, as expressed at the beginning of this section, there is no generic metric that performs well in every scenario. In [79], it was reported that sometimes the expected results using SSIM were unexpected and non-intuitive. The index computes the perceptual similarity between the ground truth images and the translated images. SSIM calculates image degradation by considering perceived changes in structural information such as luminance, contrast, and texture factors.

Inception score (IS) IS is another score created to measure image quality through a GAN model. Salimans proposed it and leveraged an Inception v3 Network pre-trained on ImageNet to determine a statistic of network outputs when applied to generated images [80]. The score ranges in $[0; +\inf[$ and it was invented to express the image quality taking into account two main factors: (1) quality: this factor is mainly expressed in terms of how realistic a generated image is; (2) diversity: the entropy level in a generative model should be high enough.

Fully Convolutional Networks score (FCN-score) The FCN score is mainly used in translating semantic maps to real photos. It is calculated using a pre-trained FCN model that evaluates image qualities by segmenting the generated images and comparing them with the ground truth label [67].

–3–
Image quality enhancement for iris segmentation

As illustrated in previous sections, state-of-the-art applications for iris recognition have mostly preferred using NIR cameras to achieve high-quality performance. However, this thesis proposes an alternative approach to exploit the NIR spectrum; rather than operating in the NIR spectrum directly (with cameras that acquire iris images under near-infrared light), the purpose is to discern and transfer the best properties that empower NIR images for iris segmentation to VIS images, thus boosting VIS-based recognition systems.

Therefore, this chapter is wholly centered on proposing a transcoding model to enhance the accuracy of iris segmentation by translating iris images from the VIS spectrum to the NIR spectrum. This process known as Image-to-Image translation (I2I) involves mapping VIS iris images into the NIR domain, preserving the intrinsic content of the original domain but with the extrinsic style of the target domain. In this regard, the concept is to transfer the advantageous characteristics and the inherent style of the NIR spectrum to VIS images to improve the performance of iris segmentation.

3.1 Problem statement

The initial chapter introduces the overall structure of an iris recognition system, highlighting the segmentation phase as a pivotal step in the process. The iris, a delicate circular membrane situated between the cornea and the lens of the eye, regulates the quantity of light entering through the pupil, a circular opening in the system. The pupil's size fluctuates between 10% to 80% of the iris's diameter, which is roughly 12 *mm* in size [81]. For accurate results, a segmentation algorithm dealing with iris image analysis needs to be resilient in the face of various obstacles, which encompass: (1) pupil dilation and constriction: changes in illumination conditions cause the pupil to dilate or constrict,

affecting the shape and size of the iris; (2) reflections and glare: external light sources or eyeglasses can produce unexpected reflections or glare, which can significantly affect segmentation accuracy; (3) eyelashes and eyelids: the occlusion caused by these artifacts makes it challenging to recognize iris boundaries; (4) noise and artifacts: low-quality images, image noise, or various artifacts can complicate segmentation process.

(a) VIS acquisition (b) NIR acquisition

Figure 3.1: Iris images extracted from the Hong Kong Polytechnic University Cross-Spectral Iris Images Database [82].

Handling reflections and glare poses a significant challenge; those issues are more prevalent and pronounced when acquiring iris images using a VIS camera and substantially impact the performance of iris recognition systems. Consequently, researchers have increasingly favored using NIR cameras, partly due to this concern. An NIR camera captures the thermal energy emitted by the iris, thus reducing the impact of visible light sources. In the NIR domain, iris images exhibit higher contrast between the iris and sclera, making the segmentation process more streamlined. This enhanced contrast is particularly pronounced in dark-colored irises, where the contrast between the iris and pupil is also more distinct. The strategies proposed in the following section address these problems by attempting to map captured images to visible light in the infrared domain, intending to preserve the information within the image but with the advantages of the target domain.

3.2 Proposed approach

The primary objective of the proposed methodology is to elevate the quality of iris image by its representation in the NIR domain. Taking a look at the literature, the breakthrough in tackling I2I problems has been represented by GAN models, which have been involved in various fields such as image synthesis, image segmentation, or image super-resolution, becoming the main architecture that has led to brilliant results in numerous tasks. Therefore, two architectures are proposed in the following sections: the former involves a neural network inspired by U-Net [75], a well-known auto-encoder

architecture in biomedical image analysis that has achieved outstanding results in many fields of computer vision and pattern recognition, while the latter is inspired by the Pix2pix [67] software, a baseline conditional GAN image translation model.

3.2.1 Data pre-processing

A common data pre-processing technique involves adding Gaussian noise to data distribution. Several reasons have motivated this process, such as (1) regularization: Gaussian noise helps to prevent overfitting and forces the model to focus on the most significant features; (2) robustness testing: training the model with noisy data increases the likelihood of good performance in real-world situations. Gaussian noise injection into a neural network input is referred to as *random jitter*; numerous studies have pointed out the beneficial impact of injecting minor input noise into the training data for improved generalization and fault tolerance [83].

The automatic pre-processing pipeline takes in input an RGB image of size $256\times320\times3$, resizes it to a larger height and width $286 \times 350 \times 3$, and randomly crops it back to the original shape $256\times320\times3$. In addition, each input is subject to random mirroring, namely, an input can randomly be flipped horizontally. To conclude, each image is normalized in the range $[-1, 1]$.

3.2.2 U-Net-based

The enormous success achieved by all neural network architectures presented in the second chapter on I2I tasks shares a common secret: within these architectures lies a concept of an encoder-decoder pattern. As stated at the beginning of this section, the first architecture consists of a U-shaped encoder-decoder with skip connections and residual blocks. The encoder unit learns to compress the data, obtaining new data representation made up of a reduced number of variables (features); the *latent space* provides an encoded form of the source data that retains all the main properties of an image under near-infrared illumination of size $256 \times 320 \times 3$. The decoder handles a symmetric decompression that leads to generating a near-infrared image of size $256 \times 320 \times 3$ that enables mapping between two domains.

Down-sampling step

The main step in the contracting path involves three operations: a 4×4 convolution with padding and stride of 2, a batch normalization operation, and a ReLU (Rectified Linear Unit) activation function. With this configuration, the down-sampling step outputs feature maps with halved size; indeed, the number of feature maps depends on the number

of filters involved in the convolutions. The choice of a non-linear activation function aids in improving the generalization of the training data.

Up-sampling step

Otherwise, in the expansive path, the main step is made up of a 4×4 transposed convolution or *up-convolution* with padding and stride of 2, a batch normalization layer, a dropout layer (applied only to the first up-sampling step in the principal architecture) and a ReLU activation function. With this setup, the up-sampling step outputs feature maps with doubled size; as for the down-sampling step, the number of filters and the choice of the activation function align with the logic previously explained.

Generator architecture

Figure 3.2 illustrates the general architecture of the encoder-decoder, where it can be appreciated the symmetry between the down-sampling (red arrows) and up-sampling (green arrows) steps and the reason this type of architecture is so powerful. The trick that makes this architecture so powerful is the skip connections (gray arrows). They connect the down-sampling steps with the up-sampling steps to add spatial information in the deepening layers and achieve more accurate results in segmentation and classification tasks [84]. Thus, a VIS image (yellow layer) feeds into the network and is compressed into latent space by the encoder module; next, the decoder reconstructs the image by reversing the process, leading to the completion of the mapping by synthesizing a NIR image (purple layer). The last step (blue arrow) differs from the classic down-sampling step previously mentioned in its activation function; a Tanh function is applied to produce 3 channels ranging in $[-1, 1]$ as the input image.

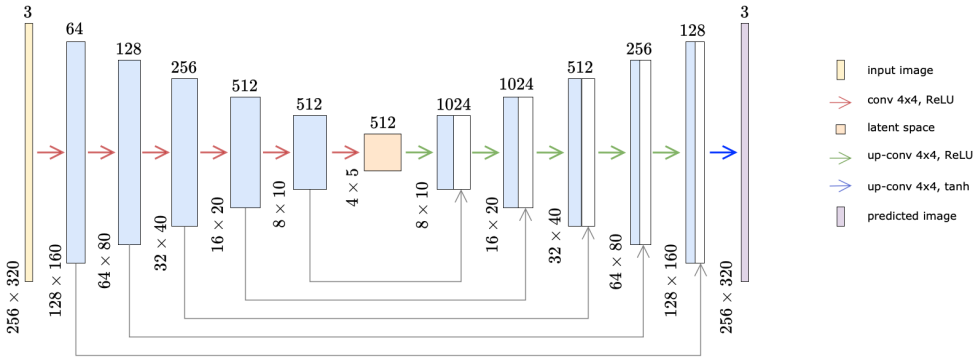


Figure 3.2: U-Net-based architecture.

Loss function

Several functions have been proposed in the literature to assess how far the predicted/-generated images deviate from ground truth/target. It is vital to choose the correct loss function for the model to minimize. Among all state-of-the-art losses related to image processing, three pixel-by-pixel metrics are selected to model the architecture (see Figure 2.3.1): Mean-Absolute Error (MAE), Mean-Square Error (MSE), Structural Similarity Index (SSIM). Despite the excellent results obtained with pixel-wise loss functions, a different strategy, patented by Johnson et al., has become popular for comparing images [85]. Instead of considering each pair of pixels in the images, which could lead to a considerable error value when, for instance, the images have a different resolution or are shifted by one pixel, perceptual loss evaluates the perceptual and semantic level between images. With a pre-trained convolutional neural network classifier, such as VGG-19, it is feasible to extract low-level features from both images (colors, lines, edges) or medium/high-level features (shapes, gradients) and analyze them to express the distance between images.

3.2.3 Pix2pix-based

The cGAN architectures have been vigorously appreciated and seem to be very effective in many tasks [5]. The zero-sum game, which the generator and discriminator establish, leads to complex results that can be advantageous for addressing specific problems and meeting expectations in most cases.

General architecture

In this section, a specific Pix2pix-based architecture is tailored to address the domain translation problem.

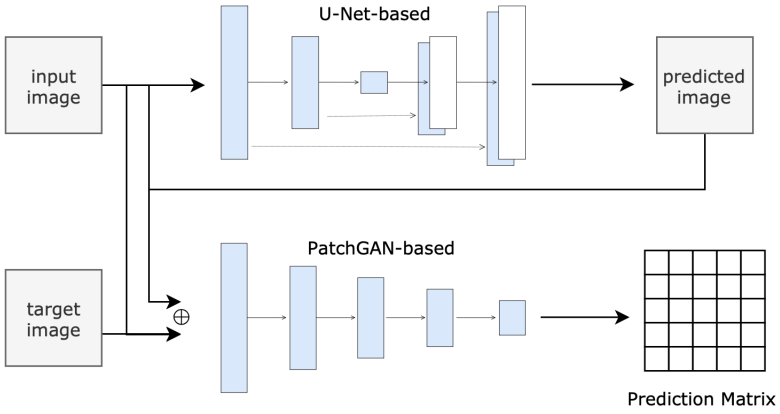


Figure 3.3: Pix2pix-based architecture.

Compared to the previous neural network, Figure 3.3 illustrates two convolutional neural networks (CNNs) combined: the generator and the discriminator. The way for generating synthetic images is unchanged; the generator in Figure 3.3 is exactly equal to the generator shown in Figure 3.2 and covers the rule of the encoder-decoder architecture. Thus, the input image remains a VIS image of size $256 \times 320 \times 3$, just like the predicted NIR image. The pivotal advancement in I2I tasks can be attributed to the discriminator. In this context, the discriminator is trained to distinguish between genuine and synthetic images, while the generator endeavors to produce synthetic images that are plausible.

Discriminator module

Many image generation jobs suffer from blurry results, which can happen when training focuses only on low-frequency structure modeling. However, low frequencies represent the core of the information, so they are nonetheless used to complete these tasks. This flaw was solved by Isola et al. who designed a discriminator that models high frequencies and analyzes the image for local patches; they named it "PatchGAN" and "penalizes only structure at the patch scale" [67].

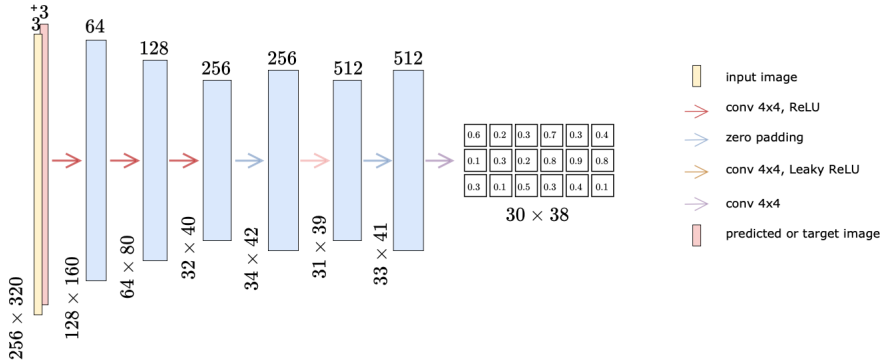


Figure 3.4: PatchGAN-based classifier architecture.

A PatchGAN-based discriminator, shown in the figure 3.4, is fed by a concatenation of two images: [*input_image*, *predicted_image*] or [*input_image*, *target_image*]. In each case, the input has a fixed shape of $256 \times 320 \times 6$, thus with double the number of channels than the generator. After feeding the network, the input undergoes three down-sampling steps (red arrows), two zero padding operations (light blue arrows), a step similar to down-sampling but with a LeakyReLU as the activation function (pink arrow), and a final 4×4 convolution with a stride of 1. The network produces 30×38 patches labeled as true or false. As a convolutional neural network classifier, its output indicates the likelihood that the image patches are real or artificially created [77]. The discriminator can accurately identify an image from a real distribution or generator network output by paying attention to local features combined with global structure.

Loss Function

The definition of the loss function in a GAN model is a very critical aspect. Since the architecture consists of two neural networks, it provides two loss functions, one to train the generator and one to train the discriminator. These two losses together form the adversarial loss function:

$$\begin{aligned}\mathcal{L}_{cGAN}(G, D) &= \mathcal{L}_{real\_dist}(D) + \mathcal{L}_{fake\_dist}(G, D) \\ &= \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))]\end{aligned} \tag{3.1}$$

The adversarial loss function measures the difference between the distribution of the produced data and the distribution of the real data [86]. During training, G is the generator that tries to minimize the objective loss function, while D is the discriminator that tries to maximize it, x is the input image, y is the truth image, and z is the random noise vector. The random noise vector z plays an important role in training because, without it, the neural network might be able to learn a direct relation between the input variable x and the output variable y, but it could only have deterministic outputs.

However, the generator G can only affect the distribution of fake data, thus its loss function is:

$$\mathcal{L}_{gen}(G, D) = \mathcal{L}_{fake\_dist}(G, D) + \lambda_P \mathcal{L}_{pw}(G) \tag{3.2}$$

Numerous approaches have found it advantageous to mix traditional loss with the adversarial loss function [67]. $\mathcal{L}_{pw}(G)$ represents the function that measures the similarity between the generated images and the ground truth images (see Section 3.2.2). Their combination leads the generator to weigh its work through λ_P, which must be to fool the discriminator but also to approach the ground truth representation.

–4–
Image quality enhancement for feature extraction

Iris recognition achieves high accuracy rates owing to the unique and stable textures of the iris. NIR cameras effectively capture detailed iris information, such as crypts, furrows, and other features, making the iris texture more distinctive. The following sections explore the relation between NIR and VIS spectra and introduce a new strategy for improving image quality, leading to the extraction of high-quality features from normalized (segmented) VIS iris images. In chapter 3, the transcoding model aims to translate VIS iris images into the NIR domain that results in more precise segmentation of irises, whereas, in this chapter, the transcoding model exploits I2I translation to enrich the acquired iris texture under VIS light. Fundamental is the development of a special loss function that evaluates the normalized NIR-like iris image transformation by incorporating the quality of the extracted features from the synthesized images through a Daugman-based algorithm. This encourages the process of producing more detailed iris textures. In the last section, a lightweight architecture is proposed to extract high-quality features from normalized VIS iris images directly, bypassing the iris image generation. It enables a comparison between indirect (through the transcoding model) and direct (through the feature extractor) iris recognition results.

In conclusion, the former approach proposes a deep learning model aiming to synthesize normalized NIR-like iris images fed by their acquisitions under VIS light, where the Daugman-based algorithm acts as a loss function during the training to encourage more representative iris textures. In contrast, the latter approach proposes a deep learning model that directly extract more discriminative features from normalized VIS iris images, trained by observing their high-quality image feature representation in the NIR spectrum.

4.1 Problem statement

As with iris segmentation, feature extraction is a thorny step in the pipeline; not surprisingly, this aspect has been extensively studied in the literature. The previous chapter outlines all the challenges associated with iris segmentation, with advantages and disadvantages of using VIS and NIR spectrum. In brief, the NIR spectrum has two key properties for the segmentation task: an increased contrast between the iris and the sclera and a reduced impact of reflections and glare. In addition, there are more motivations for researchers to operate with the NIR spectrum in the feature extraction step. A NIR camera acquires images at $700 \sim 900$ *nm* of the wavelength range leveraging "the NIR light absorption characteristics of the pigment melanin within the iris tissue which determines the visibility of iris texture details" [4]. Under VIS illumination, at $400 \sim 700$ *nm* wavelength range, iris textures appear less detailed, making textural information extraction hard.

(a) Normalized VIS iris image (b) Normalized NIR iris image

Figure 4.1: Normalized iris images extracted from the Hong Kong Polytechnic University Cross-Spectral Iris Images Database [82].

In essence, near-infrared light enables the device to acquire the complex structure of the iris region rather than its pigmentation. Therefore, being less prone to noise, NIR cameras obtain superb acquisitions even on dark-colored irises, while they remain the Achilles' heel of VIS devices [11]. In recent years, a common practice to achieve higher performance in iris recognition tasks has been to use both types of acquisition, exploiting near-infrared and visible lights. Experimental results have proven that VIS and NIR images provide complementary features for the iris pattern [87].

The proposed approaches treat all subjects with dark-colored irises expressed in polar coordinates and acquired in VIS illumination to enhance iris recognition performance.

4.2 Log-Gabor filters

The generation of an iris template results from extracting the most discriminating features, including removing unrelated and redundant data and reducing dimensionality. From the viewpoint of texture analysis, local spatial patterns of the iris are mainly made up of frequency information (number of occurrences) and orientation information (the direction in which these patterns are oriented). However, in the iris normalization process, the circular region of the iris is transformed into a rectangular region by mapping each point

from Cartesian coordinates to polar coordinates [37]. Therefore, iris details spread along the radial direction in the original representation are spread in the vertical direction in the normalized image [88]. This implies that frequency information is favored over orientation information to reflect local iris structures.

In the literature, one of the most popular and classic approaches to capturing iris texture information has been Gabor filters because they capture both frequency (as orientation and scale) and spatial (as edges and texture) information in an image. In addition, Gabor filters are remarkably effective at representing features at different scales and orientations. A Gabor filter is a linear filter "whose impulse response is given by multiplying a harmonic function with a Gaussian function" [89]. Nevertheless, Gabor filters revealed two main shortcomings over time: "the maximum bandwidth is limited to approximately one octave, and they are not optimal if one is seeking broad spectral information with maximal spatial localization" [90].

Field proposed Log-Gabor filters, which are better suited to capture information on a wide range of scales, emphasizing features at different scales due to their logarithmic frequency distribution [91]:

$$G(f) = exp\{-0.5 \times \log(f/f_0)^2 \ / \ log(\sigma/f_0)^2\} \tag{4.1}$$

where f_0 is the center of frequency and σ is the bandwidth of the filter. The convolution of a Gabor filter at different frequencies and positions over an image outputs a complex response. This complex representation allows the calculation of phase and magnitude information in the analyzed signals. However, only the phase is noteworthy since it provides more robust information than magnitude:

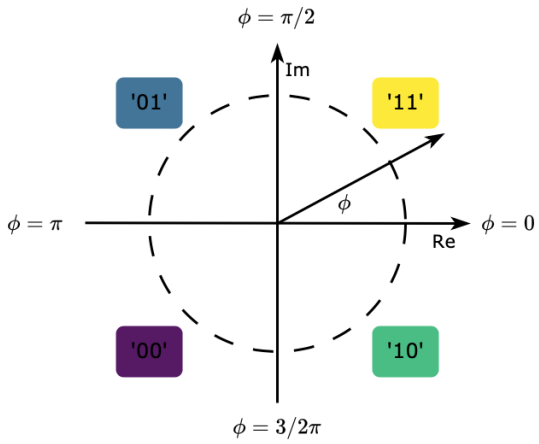


Figure 4.2: Phase-quadrant demodulation.

The phase angle can range from 0 to 2π, and phase-quadrant demodulation extracts the phase information from the complex signal by dividing the range into four quadrants (see

Figure 4.2). Then, the complex value is assigned to one of these quadrants; for instance, a complex number is expressed as:

$$z = a + bi \tag{4.2}$$

If both real and imaginary parts are positive, the pixel is assigned to the first quadrant, encoded as $'11'$ in binary representation. Applying a filter over the entire image results in a binary sequence that creates the "iris code."

Furthermore, the phase information returned by the algorithm can be represented through an image in which each pixel is the result of applying the Log-Gabor filter to that pixel in the original image and ranges in $\{0, 1, 2, 3\}$; these labels refer to the quadrant to which the angle ϕ belongs. A graphical representation is obtained by assigning a color to each label:

(a) Normalized NIR iris image (b) NIR feature image

Figure 4.3: Feature image extracted from: *151_R_NIR_3.tiff* [82].

4.3 Proposed approach

As described at the beginning of the chapter, the first approach intends to map VIS iris images, represented in polar coordinates, to the NIR domain. The researchers have proven that the irises acquired under VIS light are generally less distinct than those acquired in the NIR spectrum, particularly for dark-colored irises. Therefore, the purpose is to develop novel deep-learning models that can generate realistic NIR-like iris images regarding iris texture quality (hence, extracted features). Thus, besides applying the classical I2I translation that synthesizes iris images in the target domain mainly using pixel-wise loss, as discussed in Section 3.2, this approach proposes a new loss function that encourages the models to generate images with more detailed iris textures to extract high-quality features.

4.3.1 Daugman-based algorithm

In 1993, Daugman's study proved that the iris is biometric because, probabilistically, it is improbable for two individuals to have the same iris pattern [92]. The uniqueness of iris patterns, characterized by features such as crypts and furrows, makes the likelihood

of a random match between two individuals' iris patterns extremely low; thus, those properties make the iris reliable and secure as biometric.

In his study, Daugman applied Gabor filters to catch iris meaningful information; Gabor wavelets are effective in capturing and highlighting the unique features of the iris. The features extracted from the iris pattern presented in the image offer a different representation. The more representative and distinctive the iris texture in the image, the more informative and discriminative the extracted features tend to be. To generate an image similar to a target image, the extracted features can be used to express the differences between images, in addition to a pixel-wise comparison. The model's loss on each sample can be measured using a metric that compares extracted features between predicted images and target images. In the training process, a comparison between synthesized and original NIR-domain images occurs through loss computation.

In this work, the target images are normalized NIR iris images that are assumed to offer a higher representation of iris features. After each training step, the Log-Gabor-filter-based algorithm is used to extract features from both the real and synthesized images in the NIR domain. Section 4.2 shows that only the phase information response of the Log-Gabor filter is relevant, which can be graphically represented by an image, as shown in Figure 4.3b. However, since using a Log-Gabor-filter-based algorithm for feature extraction is computationally complex, a Convolutional Neural Network (CNN) classifier called LogGaborNet is built to approximate its behavior. This substitution is intended to improve computational efficiency and streamline the learning of the general architecture.

Data normalization

A pre-processing step prepares the data before feeding it to the neural network. Normalization, or z-score normalization, transforms each image by subtracting the mean and dividing it by the standard deviation. This process helps speed up convergence and makes the networks less sensitive to different scales. Each normalized NIR iris image of size $64 \times 240 \times 1$ undergoes the pre-processing step. Normalization can produce values out of the $[-1, 1]$ range if pixels are far from the mean in terms of standard deviation. Therefore, a final clipping is performed to ensure the values are within the correct range.

LogGaborNet architecture

The Daugam-based algorithm (see Section 4.2 takes as input a normalized NIR iris image. It produces as output a new image representation in which each pixel assumes a label in $\{0, 1, 2, 3\}$, depending on the quadrant to which the phase angle ϕ of the complex signal response belongs. Therefore, the input and output share the exact size of $240 \times 64 \times 1$. The algorithm is approximated through a convolutional neural network inspired by U-Net

[75] called LogGaborNet. The U-shaped encoder-decoder architecture compresses the input into a vector and reconstructs it into another representation. However, a precise loss function must be implemented to obtain pixel-wise output classification. Figure 4.4 illustrates the architecture of the neural network; the sequence of red arrows highlights the contracting path. Each red arrow represents the succession of three operations: a 4×4 convolution with padding, 2 stride, a batch normalization operation, and a ReLU activation function. With four down-sampling steps, the bottleneck layer shapes a feature volume of size $4 \times 15 \times 512$. Each step halves the size of the feature maps, while their number depends strictly on the number of kernels involved in each layer.

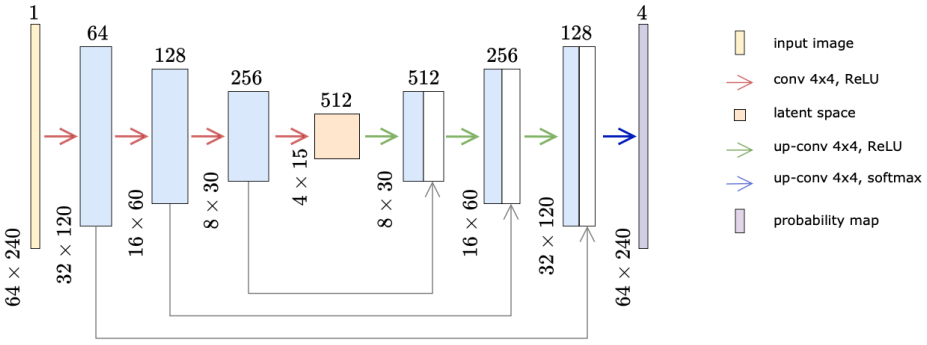


Figure 4.4: U-Net-based architecture.

Latent space provides a lower encoded input representation, retaining only the main information. The mapping process leads to extracting key features and capturing the essential characteristics of the image. Subsequently, the expansive path in the original U-Net architecture [75], is marked by the green arrows. Each step consists of a 4×4 transposed convolution or *up-convolution* with padding and stride of 2, a batch normalization layer, a dropout layer (applied only to the first up-sampling step), and a ReLU activation function. Each feature map produced by the up-sampling step, which is double the input size in the step, is joined with some skip connections from its symmetric step in the constructive path. It aids in adding spatial information in the deepening layers, and researchers demonstrated that it significantly increases performance in segmentation and classification tasks. The blue arrow represents a final convolution of 4×4 with four filters to obtain a final result of $64 \times 240 \times 4$ through the softmax activation function. It is applied to the logits of the final layer, which are real-valued numbers. The softmax function then converts these values into probabilities. In this way, each pixel in the input image corresponds to a vector of size 4, where each value is expressed in the range $[0, 1]$ and can be interpreted as a probability.

Then, each pixel is mapped to a probability distribution $\mathcal{P} = (p_0, p_1, p_2, p_3)$ where each p_i represents the probability that the pixel belongs to the quadrant i. The argmax function

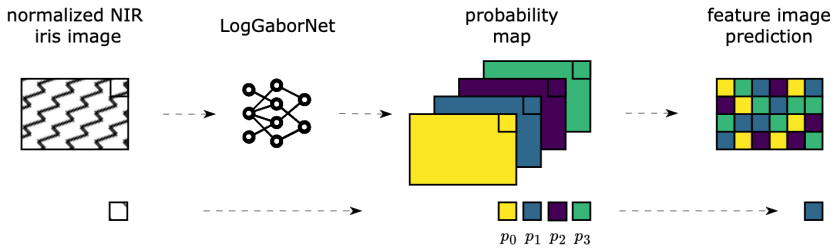


Figure 4.5: Creation of feature images.

applied to every pixel yields the prediction of the feature image.

Loss function

In a multi-class classification task, with a softmax as the activation function of the last layer, the categorical cross-entropy loss function is a common choice. The categorical cross-entropy loss compares the probability distribution of each pixel with the true distribution expressed by the one-hot encoding representation. For instance, a pixel labeled as 3 (thus belonging to the third quadrant) in the original feature image is expressed as $(0, 0, 1, 0)$ through one-hot encoding. Other types of loss functions that may be suitable for this task are involved in image segmentation. They are mainly used when the spatial arrangement of pixels is relevant. The most well-known losses in this regard are the Dice loss, also known as the Sørensen-Dice coefficient, which measures the spatial similarity or overlap between the predicted segmentation mask and the ground truth mask, and the Intersection over Union (IoU) loss, also known as the Jaccard loss, which measures the ratio of the intersection area to the union area between the predicted segmentation mask and the ground truth mask.

4.3.2 General architectures

The LogGaborNet previously described allows the definition of a feature-based loss function that computes the error based on how likely the predicted image and the target image are similar in terms of extracted features.

Data pre-processing

An augmentation step is undertaken to ensure optimal performance before training the models. The objective is to enrich the initial training set by generating diverse versions of similar contexts, thereby exposing the models to a broader range of training image samples. The process concerns creating four modified copies for each normalized VIS iris image by applying four specific transformations: horizontal flipping, vertical flipping,

and circular shifting. This approach helps to enhance the size and diversity of the data, minimize overfitting, and improve the generalization and robustness of the model.

In conclusion, the entire dataset is normalized using the same process described in Section 4.3.1, which scales the data within the $[-1, 1]$ range.

U-Net-based

The U-Net-based architecture closely resembles the LogGaborNet architecture, except for the final step (blue arrow). In that architecture (see Figure 4.4), the goal is to produce a probability map to have for each pixel a probability distribution over the four quadrants to replicate the behavior of the feature extractor algorithm described in Section 3.2. In contrast, in this architecture (see Figure 4.6) the output is an image, the representation of the input into the target domain. To achieve this, the neural network must function as an encoder-decoder, learning how to represent the normalized VIS iris images in a lower dimensional space to obtain a compressed and meaningful representation of them. Subsequently, the decompression must transfer this information into the target domain, generating normalized NIR-like iris images.

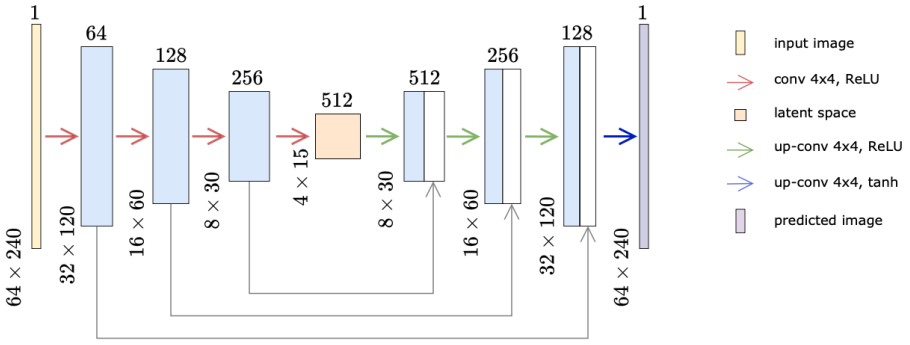


Figure 4.6: U-Net-based architecture.

The down-sampling process comprises four distinct steps, identified by red arrows. In each step, a 4×4 convolution with padding and a stride of 2 is applied, followed by batch normalization and a ReLU activation function. The input is then compressed into a latent space, which undergoes four steps in the expansive path. The first three steps, marked by green arrows, repeat the same operations as the down-sampling process: 4×4 transposed convolution with padding and a stride of 2, dropout layer (only applied to the first up-sampling step), and ReLU activation function. The last step, marked by a blue arrow, refers to a 4×4 transposed convolution with just one filter to obtain the same shape as the input $64 \times 240 \times 1$, followed by a Tanh activation function, which produces values in $[-1, 1]$.

Loss function To fine-tune the model's parameter during the training step, the following formula expresses the loss function involved in the process:

$$\mathcal{L}_{ubm}(U, F) = \mathcal{L}_{pw}(U) + \lambda_F \mathcal{L}_{fb}(U, F) \tag{4.3}$$

where $\mathcal{L}_{pw}(U)$ is determined using a pixel-wise loss function, such as mean absolute error (MAE) or mean squared error (MSE), which compares predicted normalized NIR iris images outputted by the model U and their original representation in the NIR domain. On the other hand, $\mathcal{L}_{fb}(U, F)$ is a feature-based loss function, such as Dice Loss or IoU Loss, which compares the feature extracted from images synthesized by U and those extracted by the target images, through a trained LogGaborNet model F (see Section 4.3.1). The contribution of the latter is balanced using λ_F to achieve optimal tuning.

Pix2pix-based

In contemporary literature, conditional generative adversarial network (cGAN) architectures have been extensively employed for image-to-image (I2I) translation tasks for several reasons. Firstly, the discriminator network in cGANs encourages the generator network to produce realistic images similar to the target images. Secondly, the architecture of cGANs, which involves two convolutional neural networks (CNNs), enables them to capture complex relations between input and target domains that may otherwise be difficult to model. Finally, as one of the most groundbreaking discoveries in machine learning, the development and refinement of GAN-based architectures have resulted from continuous community efforts to improve their performance. As for the previous task, the architecture is inspired by Pix2pix [67], a cGAN model for image translation problems. The previous U-Net-based architecture (see Figure 4.6) acts as the generator module in this Pix2pix-based architecture. Its purpose is to synthesize normalized NIR-like iris images, covering the encoder-decoder (see Figure 4.7) powered by the discriminator module, which is trained to distinguish between real and synthetic images. During the training, the generator's task is to synthesize images that are as realistic as possible to fool the discriminator, making its job difficult. As discriminator architecture, Isola et al. [67] proposed a Patch-GAN discriminator to discriminate images from local patches rather than the whole image. The researchers introduced it to reduce the blurring of the generated images since many works have addressed this problem; this occurs especially when the generator loss function consists of per-pixel losses, such as L1 or L2. However, this is quite normal and necessary to preserve the low frequencies, which contain the core of information. The Patch-GAN discriminator takes care of the high frequencies resulting in more shaped and detailed image generation.

The generator, as explained after in Eq.4.5, is trained by the union of three losses that measure three fundamental parameters: the first loss measures the error made by the discriminator in evaluating synthetic images as real, the second loss (a per-pixel loss) helps

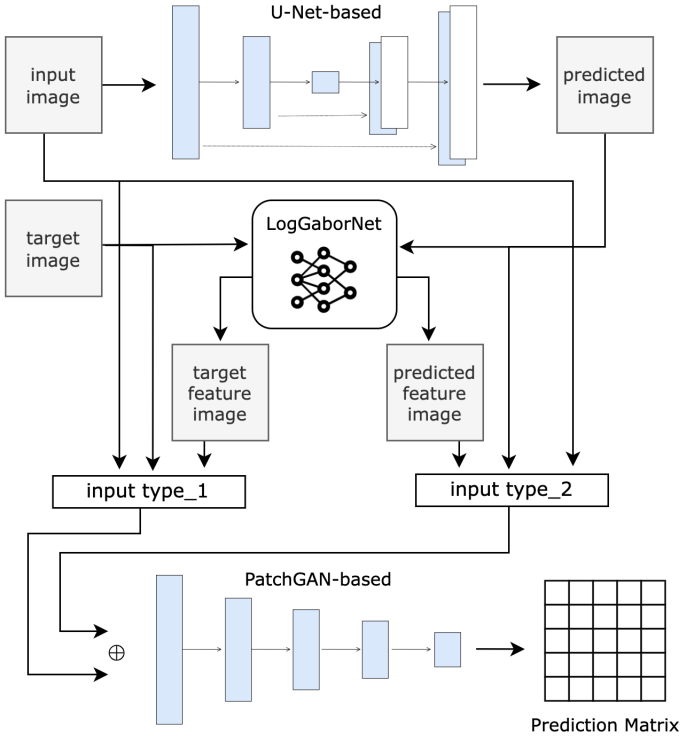


Figure 4.7: Pix2pix-based architecture.

maintain structural consistency and fidelity the synthesized images to target images while the third loss (a feature-based loss) encourages the generation of normalized NIR-like iris images penalizing differences in extracted features between synthesized images and target images. The purpose of the generator is to produce synthetic images that closely resemble the target images, even in terms of extracted features. The more similar the iris images are in terms of features, the more distinct the features of the synthetic image become, resulting in a more accurate representation of the iris pattern and an overall improvement in image quality. The discriminator must be able to differentiate between the distribution of fake and genuine iris images by evaluating feature images, which then influences model prediction based on feature quality. Figure 4.8 illustrates its architecture; it captures fine-grained details and spatial relationships within the image. The model can assess the image's realism as each patch is treated independently during the training. It requires three input images: [*input_image*, *predicted_image*, *predicted_feature_image*] or [*input_image*, *target_image*, *target_feature_image*] of total size $64 \times 240 \times 3$ represented by the yellow layer, the red layer and the gray layer, respectively.

The input undergoes three steps (red arrows): a 4×4 convolution with padding and stride of 2, a batch normalization operation, and a ReLU activation function. Look at

Figure 4.8; red arrows are followed by two light blue arrows, representing two operations of zero padding, then a pink arrow between them and the last violet arrow. The pink step is very similar to the red step, except for the stride, which is 1 in this step. Indeed, the last step, which outputs the classification matrix, is a 4×4 convolution with a stride of 1. The network produces 6×28 patches labeled as true or false, indicating the likelihood that the image patches are real or artificially created [77].

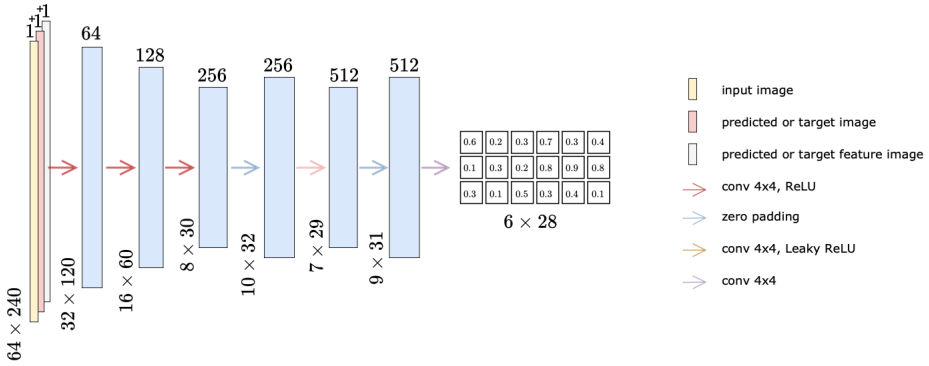


Figure 4.8: PatchGAN-based classifier architecture.

Loss function The primary objective of a cGAN model is to create synthetic data that conforms to a distribution of data conditioned on specific input information. In [70], Goodfellow et al. proposed a loss function called minimax loss which reflects the distance between the original distribution and the distribution of the data generated by the GAN. In a GAN architecture, two loss functions work together and derive from a single distance measure between probability distributions. Even in the case of cGAN, minimax loss can be employed as the output y is conditioned on an input x, resulting in a mapping $G : (x, z) \longrightarrow y$. The proposed minimax loss function for this architecture also considers the LogGaborNet F in the formula:

$$\begin{aligned} \mathcal{L}_{cGAN}(G, D, F) &= \mathcal{L}_{real\_dist}(D, F) + \mathcal{L}_{fake\_dist}(G, D, F) \\ &= \mathbb{E}_{x,y}[\log D(x, y, F(y))] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z), F(G(x, z))] \end{aligned} \tag{4.4}$$

where $D(x, y, F(y))$ denotes the probability that the sample y and its feature image $F(y)$ came from the data distribution given condition x, $G(x, z)$ is a fake generated sample conditioned on x and $F(G(x, z))$ is its feature image that comes from the LogGaborNet [93]. The discriminator is trained to minimize this function, while the generator is trained to maximize it. However, the generator can only affect the $\mathcal{L}_{fake\_dist}(G, D, F)$ term in the formula because it reflects the distribution of the fake data. So, the first term, which reflects the real data distribution, is dropped during generator training. Moreover, Isola et al. obtained improved results and decreased blurring in their study by adding additional

loss terms (as L1 or L2) to the loss used to trick the discriminator [67]. Thus, the proposed loss function for the generator module involves both pixel-wise (as Isola et al. suggested) and feature-based losses as in the U-Net-based architecture (see Section 4.3.2):

$$\mathcal{L}_{gen}(G, D, F) = \mathcal{L}_{fake\_dist}(G, D, F) + \lambda_P(\mathcal{L}_{pw}(G) + \lambda_F \mathcal{L}_{fb}(G, F)) \tag{4.5}$$

4.4 Direct extraction of high-quality features

The previous approach presents two tasks: the former builds a model that approximates a Daugman-based algorithm through LogGaborNet for feature extraction, while the latter incorporates the LogGaborNet model in its generator loss function leading to the synthesis of high-quality iris images in polar coordinates. The approach aims to improve the quality of images acquired under visible light by considering the iris' representativeness in the NIR spectrum and the features extracted from that spectrum. However, by simplifying the architecture, it is possible to design a simplified one that leads directly to the generation of features similar to those extracted from the irises represented in the NIR domain, thereby increasing the quality of those features. At this point, the alternative approach becomes a feature extractor where the input is still a normalized VIS iris image while the final output is a feature image. The two architectures proposed in the following sections are based on U-Net and Pix2pix architectures previously used to address a similar problem: the U-Net [75] and the Pix2pix [67] architecture.

4.4.1 Data normalization

The data is pre-processed before the training step. This process involves normalization, which transforms each image by subtracting its mean and dividing it by the standard deviation. This step aims to speed up convergence and make the networks less sensitive to different scales. The pre-processing step is implemented on each normalized VIS iris image of size $64 \times 240 \times 1$. However, normalization can sometimes produce values outside the $[-1, 1]$ range if pixels are far from the mean in standard deviation. A final clipping is performed to ensure that the values remain within the correct range.

4.4.2 U-Net-based

The figure 4.9 presents the U-shaped encoder-decoder that focuses on encoding the input representation in a latent space of size $4 \times 15 \times 512$, retaining the main information about the original input representation, such as essential features. The contracting path is made up of four down-sampling steps (red arrows). Each step represents a 4×4 convolution with padding and stride of 2, a batch normalization operation, and a ReLU activation function. The number upon each layer symbolizes the number of kernels adopted during

convolutions. It influences the number of feature maps in the next step (except for the yellow and purple layers, representing the input and output images, respectively). Later, an expansive path reconstructs the input: each up-sampling step (green arrows) consists of 4×4 transposed convolution with padding and stride of 2, a batch normalization layer, a dropout layer (applied only to the first up-sampling step) and a ReLU activation function. The reconstruction process leverages the skip connections of the encoding process to add spatial information in deepening layers, increasing the performance of the classification task.

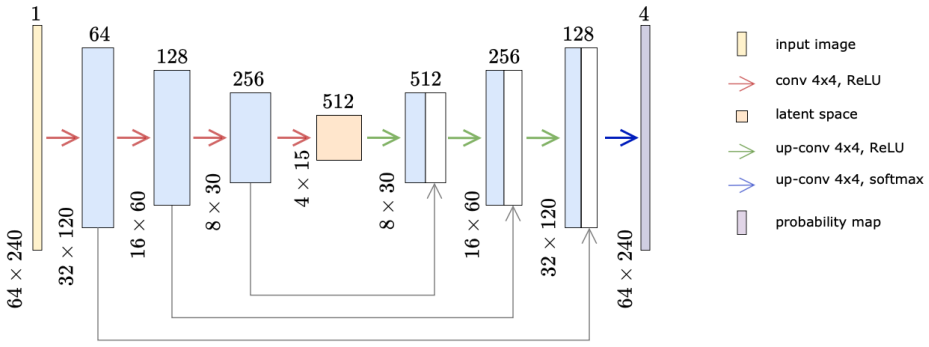


Figure 4.9: U-Net-based architecture.

The algorithm based on the Log-Gabor filter produces four possible outcomes, depending on the quadrant to which the ϕ angle phase belongs for each pixel in an image, as explained in 4.2. To replicate an output that matches the output of the feature extraction algorithm, the blue arrow in the diagram, representing the final convolution of 4×4, applies 4 filters to obtain a final result of $64 \times 240 \times 4$ with a softmax activation function. Thus, the last step produces four prediction maps, one for each quadrant. The softmax activation function converts the logits of the final layer into probabilities. This conversion produces a probability distribution. Each pixel in the image is assigned a probability distribution $\mathcal{P} = (p_0, p_1, p_2, p_3)$ where p_i represents the probability that the pixel belongs to the quadrant i. Finally, the argmax function is applied to every pixel to yield the prediction of the feature image.

4.4.3 Loss function

The input image is mapped into a probability map where each pixel is associated with a probability distribution across the possible labels (quadrants). A common choice as a loss function is the categorical cross-entropy; it measures the distance of the probability distribution of each pixel from the truth value. For instance, a pixel labeled 0 (thus belonging to the first quadrant) in the original NIR feature image is expressed as (1, 0, 0, 0)

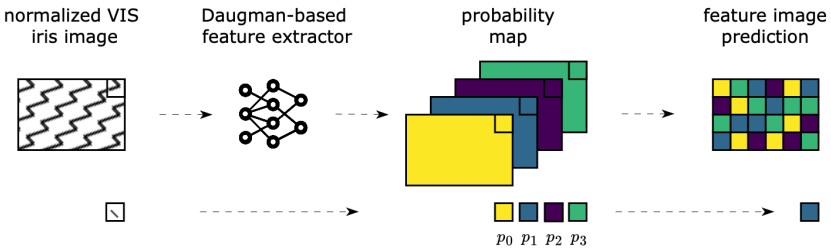


Figure 4.10: Creation of feature images.

through one-hot encoding. Other losses suitable for training the models are Dice loss, also known as the Sørensen-Dice coefficient, and the Intersection over Union (IoU) loss, also known as the Jaccard loss.

4.4.4 Pix2pix-based

The previous Pix2pix-based architectures, illustrated in Figure 3.3 and Figure 4.7, are designed to address I2I translation by attempting to represent VIS iris images in the NIR domain. Therefore, the representation of the iris as realism within the synthetic images is crucial. However, since the goal is different in this section, the target domain changes in the following architecture (see Figure 4.11). In this case, the target domain consists of all the feature images produced by applying the Daugman-based algorithm to the normalized iris images acquired in NIR light (target feature images). Therefore, the models directly map the normalized iris VIS images (input images) into the feature images (predicted feature images) extracted from their representation in the NIR spectrum.

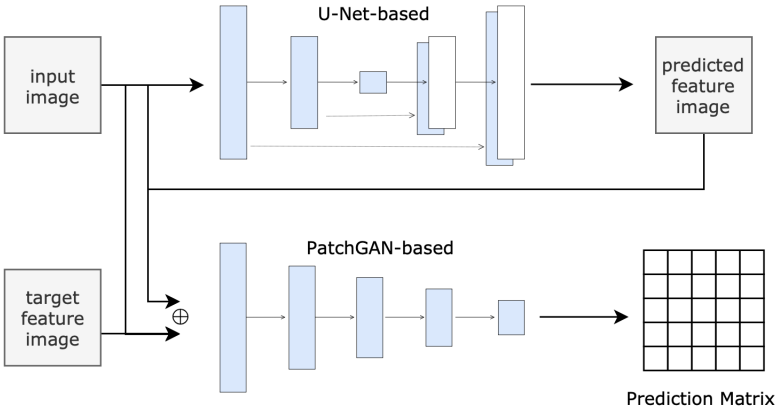


Figure 4.11: Pix2pix-based architecture.

The architecture proposed in Section 4.4.2 acts as a generator in the generator-

discriminator architecture above. In contrast, the discriminator architecture is modeled based on the PatchGAN discriminator designed by Isola et al. [67].

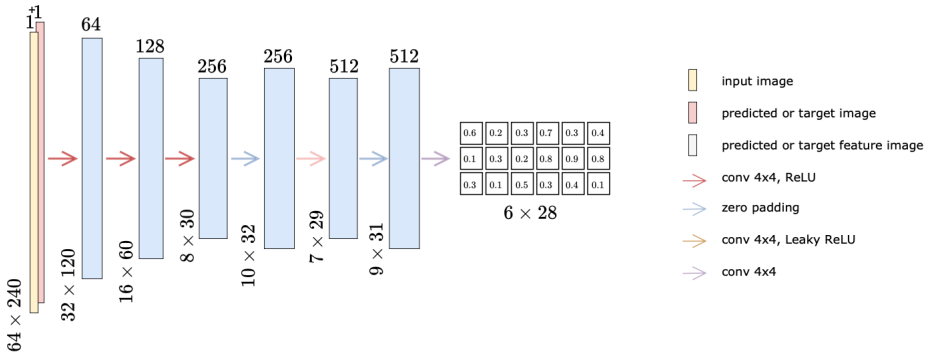


Figure 4.12: PatchGAN-based classifier architecture.

The CNN above processes the input through three down-sampling steps (see Section 3.2.2). Each step (red arrows) involves a 4×4 convolution with padding and stride of 2, followed by a batch normalization operation and a ReLU activation function after a zero-padding operation (light blue arrow) precedes a 4×4 convolution with padding and stride of 1, a batch normalization operation, and a LeakyReLU activation function (pink arrow). The last steps concern another zero-padding operation (light blue arrow) and a 4×4 convolution with padding and stride of 1 (purple arrow).

The discriminator's rule distinguishes real data distribution from fake data distribution, discriminating them from local patches. To each step, the discriminator expresses the likelihood that the input belongs to the real distribution; firstly taking in input [*input_image*, *predicted_feature_image*], and secondly taking [*input_image*, *target_feature_image*] where *predicted_feature_image* is produced as explained in Figure 4.10. Both inputs have a size of $64 \times 240 \times 2$, represented by the yellow layer and the red layer.

4.4.5 Loss function

In Goodfellow's original formulation of generative adversarial networks (GANs) [70], the minimax loss function addresses the problem of reaching an equilibrium in which the generator produces realistic samples and the discriminator is no better than a random guess. Therefore, the original formula is extended to incorporate conditioning information:

$$\begin{aligned} \mathcal{L}_{cGAN}(G, D) &= \mathcal{L}_{real\_dist}(D) + \mathcal{L}_{fake\_dist}(G, D) \\ &= \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))] \end{aligned} \tag{4.6}$$

the generator tries to minimize it by generating data that the discriminator classifies as real, thus affecting the first term $\mathcal{L}_{real\_dist}(D)$. The discriminator tries to maximize

it by distinguishing between true and false data by also considering the second term $\mathcal{L}_{fake\_dist}(G, D)$. As a multi-class classification task, the generator, in its loss function, also measures the distance of dissimilarity between the predicted probability distribution and the true distribution of the target classes through the loss presented in the Section 4.4.3:

$$\mathcal{L}_{gen}(G, D) = \mathcal{L}_{fake\_dist}(G, D) + \lambda_P \mathcal{L}_{mc}(G) \tag{4.7}$$

where the combination of the two loss functions is weighted by λ_P.

–5–
Training details and evaluation

This chapter covers all essential aspects related to training, including dataset creation and implementation details. It further includes the evaluation and presentation of results, outlining the metrics used for model selection and performance analysis. Finally, it proposes a general evaluation of the best models in terms of iris recognition.

5.1 PolyU Database

Tasks illustrated in the previous chapters entail training models to improve the quality of the VIS iris images by leveraging its representation in the NIR spectrum. As a result, the dataset must comprise pairs of images acquired simultaneously using a bi-spectral approach. The Hong Kong Polytechnic University developed a database with pixel-to-pixel correspondences between two spectral iris images to study cross-spectral iris recognition performance [82].

The folder '*PolyU_Cross_Iris*' contains 12, 540 iris images. These images are of size 640×480 and were acquired from 209 different subjects. Each subject has 15 instances for each eye (left and right) in both VIS and NIR spectra. The data for each subject is stored in a separate folder numbered from '*001*' to '*209*'. Each subject's folder contains two sub-folders, *L* and *R*, which are further divided into '*VIS*' and '*NIR*' sub-folders. Additionally, the first session includes another folder named '*PolyU_Cross_Norm_Unenhanced*', which has the same structure (and subjects) as the former but contains normalized (segmented) iris images . These images are of size 64×512. In the folder '*PolyU_Cross_Iris*', VIS iris images are represented using three color channels in the RGB (Red, Green, Blue) color space. Each channel contains pixel values corresponding to the intensity of a specific color, and the combination of these channels forms a color image. In contrast, the folder '*PolyU_Cross_Norm_Unenhanced*' contains normalized VIS iris images in a gray-scale representation. Gray-scale images have only one channel, where pixel values represent

the brightness intensity. The absence of color channels simplifies the image representation to a single channel, representing the luminance or intensity of each pixel.

5.1.1 Creation of datasets

The two macro folders led to building the two datasets involved in transcoding for the iris segmentation (see Section 3.2), transcoding for feature extraction (see Section 4.3), and the creation of the feature extractor (see Section 4.4). For both datasets, (x, y) samples are created for each subject where x represents the image acquired under the visible light while y is in the near-infrared spectrum. The datasets are split into three subsets, namely training, validation, and test sets:

	Training Set	Validation Set	Test Set
subjects	001 to 150	151 to 180	181 to 209

where the training set is used to train the model, while the validation set is used to evaluate the model performance during training and fine-tune hyperparameters. On the other hand, the test set is completely independent of the other two sets, and the model has never seen it before during training or validation. The test set is used to assess the final, unbiased performance of the trained model.

However, some adjustments are required to be fully compatible with the proposed architectures. Regarding the image quality enhancement for the segmentation step, each image in the folder '*PolyU_Cross_Iris*' is resized to size 256×320. Moreover, to facilitate model learning, all right irises are horizontally reflected to make the dataset more uniform. It ensures that the dataset presented only images with medial commissure on the left and lateral commissure on the right. Only the first four acquisitions for each eye are considered for each subject, making eight acquisitions per subject. Since the models replace the output images with the same size as the input, all synthesized images are resized to the original shape of 640×480 during model evaluation. Finally, all the right iris images are reflected in the original position.

Instead, for image quality improvement for the feature extraction stage and the creation of a method for extracting high-quality features, the same dataset extracted from the folder '*PolyU_Cross_Norm_Unenhanced*' is used in which each image was resized to 64×240.

5.2 Implementation details

The entire project was implemented using Python 3 and related libraries. The training was carried out on Kaggle, an environment that provides free GPU usage for up to 30

hours per week. To speed up the training process, the accelerator GPU P100 was selected. TensorFlow, an open-source platform, and Keras, its high-level neural network library, were instrumental in building all the deep learning models. In addition to these machine-learning tools, several libraries played a crucial role in the project's success. Numpy was explicitly designed to make scientific computing faster and more accessible. Matplotlib is a powerful data visualization tool that enabled the creation of many helpful illustrations, and OpenCV provided various functions for basic and advanced image processing tasks.

5.3 Evaluation metrics

This thesis focuses mainly on Image-to-Image (I2I) translation tasks, so quality can be assessed using two metrics: the Peak Signal-To-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM). The former is a pixel-wise metric that considers corrupting noise that affects the quality of the representation. At the same time, the latter is a more complex metric that considers perceptual aspects and structural information, providing a measure of similarity more in line with human perception.

While the two distinct transcoding tasks, both of which necessitate evaluating the images using quality metrics to ensure the fidelity of the synthetic image representations in the new domain, the primary objective of this research work is to enhance the performance of iris recognition.

In pursuit of this objective, models based on the architectures presented in Section 3.2 3 are subjected to evaluation through an iris segmentation method to determine whether these models can generate synthetic images that exhibit superior segmentation results compared to the original images captured in visible light. De Marisco et al. proposed IS_{IS}, an iris segmentation method that entails four primary steps: (1) pre-processing the images to eliminate irrelevant details, (2) applying Canny filtering, Taubin's algorithm, and a voting procedure to identify circles and localize the pupil, (3) linearizing the image in polar coordinates, and (4) locating the limbus using median filtering and weighted difference calculation to find the boundary between the iris and sclera [94]. The assessment is conducted by leveraging ground truth masks, and the outcomes are quantified through a diverse set of metrics. These metrics encompass the Dice coefficient, Jaccard Index (IoU), Precision, Recall, and F1-score (see Section 5.4.1).

Similarly, the models structured upon the architectures delineated in Section 4.3.2 are evaluated in terms of iris recognition rate to discern the contribution of the features extracted from the synthetic images. Employing the algorithm explicated in Section 4.2, a feature image is derived from each synthetic image. This feature image can be represented as an iris code, characterized by a binary sequence. Specifically, the response of the Gabor filter on a pixel is encoded in binary format, as illustrated in Figure 4.2. Next, the matching phase includes the computation of pairwise Euclidean distances for each pair of

examples, storing genuine pairs (DI) and impostor pairs (DE). The Decidability (Dec) is subsequently calculated as:

$$Dec = \frac{|(\bar{DI} - \bar{DE})|}{\sqrt{(0.5 * (\tilde{DI}^2 + \tilde{DE}^2))}} \tag{5.1}$$

where $\bar{DI}$ and $\bar{DE}$ represent the mean and $\tilde{DI}$ and $\tilde{DE}$ the standard deviation of genuine and impostor distances, respectively. Finally, a pivotal metric, the Equal Error Rate (EER), is calculated which determines the point where the False Acceptance Rate (FAR) and Genuine Acceptance Rate (GAR) are closest (see Section 5.4.2).

Chapter 4 also presents the LogGaborNet where every pixel of the input image is classified into one of four labels: $\{0, 1, 2, 3\}$ (see Section 4.3.1). To evaluate its performance, several metrics are chosen to compare synthetic and target feature images: Dice coefficient, Jaccard Index (IoU), Accuracy, and F1-score. All metrics are calculated by first computing them for each class independently, and then averaging them across all classes using a micro-averaging approach (see Section 5.4.2).

In conclusion, the last part of the thesis work, contained in Section 4.4, involves the direct implementation of a feature extractor, focusing on a per-pixel classification task. The performance is evaluated either trivially by measuring the pixel classification error in the image as Dice coefficient, Jaccard Index (IoU), Accuracy, and F1-score, or by calculating the rate of Dec and EER obtaining a measure in terms of recognition (see Section 5.4.2).

5.4 Separate evaluation of models

This section presents the models that performed best in the previously described tasks, their empirical evaluations, and visual illustrations. An automatic custom data loading function applies the pre-processing step and shuffles the data into batches. All the weights are initialized through a normal distribution with 0 mean and 0.2 standard deviation. Adam's stochastic gradient descent is chosen to optimize the training with a learning rate of $2e-4$. Different neural network hyper-parameter configurations are tested for the best performance and the perfect tuning, considering Kaggle's memory and time size limits.

5.4.1 Transcoding models for iris segmentation

For this task, named $t1$, the dataset comprises the first 4 acquisitions per eye of all the subjects, as explained in Section 5.1.1. It undergoes a pre-processing step before feeding the models (see Section 3.2.1). The training set has $1,200$ samples, the validation is 240, and the test set is 232, where each sample is a pair of 'VIS, NIR' iris images.

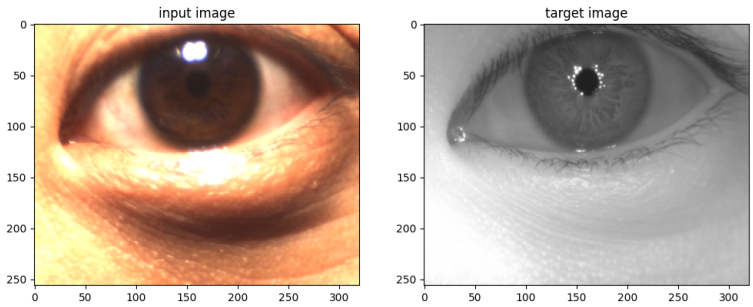


Figure 5.1: Iris image acquisition under visible and near-infrared light

Two different architectures are illustrated in Chapter 3 to address the problem: U-Net-based (UNB) and Pix2pix-based (PPB) architectures. The different external configurations involved in model initialization mainly differ from the loss functions adopted (see Section 3.2.2 and Section 3.2.3) and the batch size.

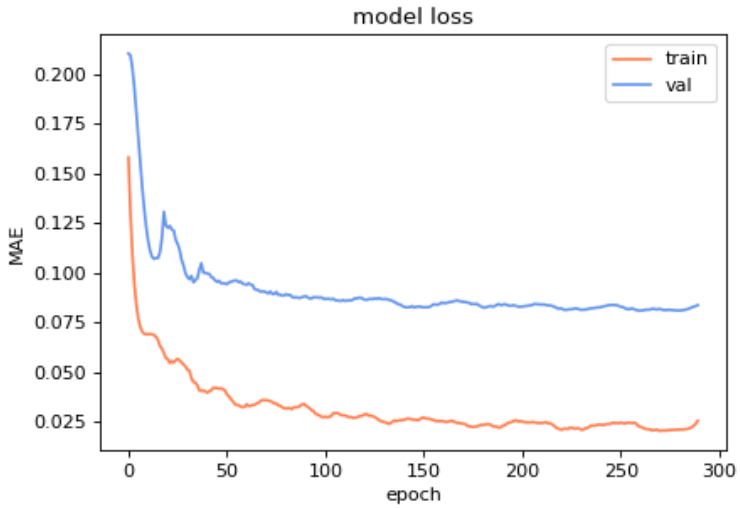


Figure 5.2: *UNB_model*$_{t1}$ at epoch 239/289; metrics on validation set: MAE : 0.0787, SSIM : 0.87 ± 0.01 , PSNR : 26.34 ± 13.05.

In this regard, the model selection declares that the best model on the validation set is the *UNB_model*$_{t1}$ with a batch size of 4 and the L1 loss function. The training lasts for 289 epochs with an average run time of 65 seconds per epoch on GPU P100. The process is interrupted by the early stopping regularization technique with patience of 50. Figure 5.2 shows that over-fitting is not noticed for the whole training, meaning that data augmentation and other regularization techniques (such as dropout and early stopping) are effective. Beyond the Mean Absolute Error (MAE), which essentially measures the difference between synthetic and target images in terms of pixel values to provide a sense

of overall fidelity of synthetic images, other metrics used to monitor the training: the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM).

To have a comprehensive understanding of the experiment, the best PPB model is compared to the selected model (see Figure 5.2) and its performance is analyzed in detail, highlighting its limitations and shortcomings. The best configuration is obtained setting $\mathcal{L}_{pw}(G)$ as VVG loss function with $\lambda_P = 1000$ (see Section 3.2.3), and a batch size of 4 to maximize the performance during the training. Table 5.1 illustrates the evaluation metrics for the two models: the UNB model performs better than the PPB model across all three metrics (lower MAE, higher SSIM, and higher PSNR); it implies that the UNB model is more effective in producing synthesized images close to their real representations either in terms of structural contents and in terms of pixel values (fidelity aspects). Figure 5.3, Figure 5.4, and Figure 5.5 showcase the inference results of the models on selected examples from the validation set. The PPB model exhibits satisfactory performance under standard conditions, yielding predictions that exhibit a heightened level of realism (see Figure 5.3) compared to its UNB counterpart.

	MAE	SSIM	PSNR
UNB_model_{t1}	**0.0787**	**0.87 ± 0.01**	**26.34 ± 13.05**
PPB_model_{t1}	0.1123	0.81 ± 0.01	23.29 ± 11.06

Table 5.1: Comparison of U-Net-based and Pix2pix-based models on the validation set.

However, the former model manifests reduced robustness in noise as iris deformation, fluctuations in illumination, reflections near the pupil boundary, or deviations from the optimal gaze angle. As evidenced by Figure 5.5, a notable shortcoming emerges when the model encounters challenges in accurately localizing the pupil. In such instances, the PPB model attempts to reconstruct the pupil during translation, drawing from its learned domain experiences. However, there are occurrences where the PPB model fails to predict the pupil adequately, significantly affecting the subsequent iris segmentation step. Accurate pupil reconstruction is imperative in this step for precisely extracting the iris from the image.

	MAE	SSIM	PSNR
UNB_model_{t1}	**0.0787**	**0.87 ± 0.01**	**25.03 ± 22.34**

Table 5.2: Evaluation of the selected model on the test set.

UNB_model$_{t1}$ is more appropriate for the task at hand. Therefore, it undergoes a final test set evaluation to provide a more accurate estimate of completely unseen data to estimate the model performance in real-world scenarios. The model processes images in a way that is less prone to glare and reflections, has better control over illumination conditions, which is crucial for maintaining consistent image quality in different acquisition environments, and better differentiation between the iris and surrounding structures, such as the sclera and pupil.

Furthermore, *UNB_model*$_{t1}$ is tested in terms of iris segmentation using the IS_{IS} technique (see section 5.3). Since the Poly's database does not provide ground truth iris masks, the experiment involves the MICHE dataset [10], which contains VIS iris images captured under uncontrolled settings using mobile devices to simulate real-world scenario acquisitions. The IS_{IS} technique provides iris masks for all original VIS iris images in the dataset and all NIR-like iris images resulting by *UNB_model*$_{t1}$. In addition, the evaluation includes the performance of the segmentation method employed in Watershed-based iris recognition (WIRE) applied to the original MICHE images [95]. This method stands out for its innovation, particularly in addressing a pre-processing step specifically for dark-colored irises, and is considered a state-of-the-art technique. Successively, the iris masks are then compared to precise manual segmentation to measure the iris segmentation performance.

	Dice	IoU	Precision	Recall	F1-score
IS_{IS}*-UNB masks*	0.95 ± 0.04	0.90 ± 0.07	0.96 ± 0.04	0.94 ± 0.06	0.95 ± 0.04
IS_{IS} *masks*	0.96 ± 0.05	0.92 ± 0.08	**0.97 ± 0.03**	0.95 ± 0.08	0.96 ± 0.05
WIRE masks	**0.97 ± 0.04**	**0.94 ± 0.07**	0.97 ± 0.04	**0.95 ± 0.06**	**0.97 ± 0.04**

Table 5.3: Iris segmentation results with different methods on the MICHE dataset.

The information presented in Table 5.3 indicates that *UNB_model*$_{t1}$ does not improve image quality, leading to better results in iris segmentation and suggesting that the PolyU database alone does not make the model robust across different datasets. The MICHE dataset contains several types of image acquisitions that include specular reflections, blurring, distortion, and other noise factors that are not present in the PolyU database (see Section 5.1) used to train the model.

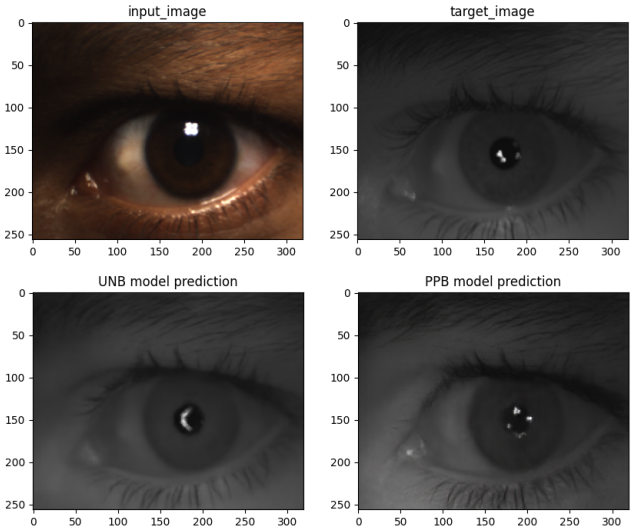


Figure 5.3: Predictions of UNB_model_{t1} and PPB_model_{t1} on subject 151, left eye, first acquisition.

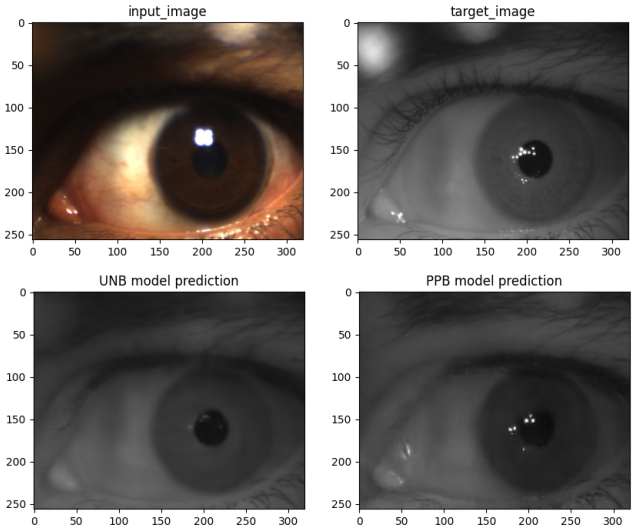


Figure 5.4: Predictions of UNB_model_{t1} and PPB_model_{t1} on subject 153, right eye, first acquisition.

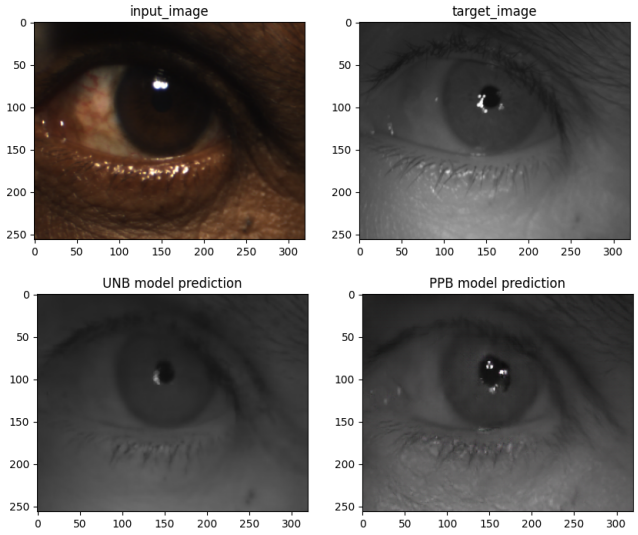


Figure 5.5: Predictions of UNB_model_{t1} and PPB_model_{t1} on subject 167, left eye, forth acquisition.

5.4.2 Transcoding models for feature extraction

This task, named $t2$, involved the training and the selection of two models: the former is a feature extractor called LogGaborNet that approximates the behavior of a Daugman-based algorithm while the latter integrates the mentioned model in its loss function for image transcoding.

LogGaborNet The dataset comprises only normalized NIR iris images following the same division into sets as expressed in Section 5.1.1. The division comprises 4, 500 samples for the training set, 900 samples for the validation set, and 870 for the test set, where each normalized NIR image is paired with its feature image extracted applying the Daugman-based algorithm (see Section 4.2) which act as ground truth during the training. The figure below shows an image picked from the training set after the pre-processing step and its feature image.

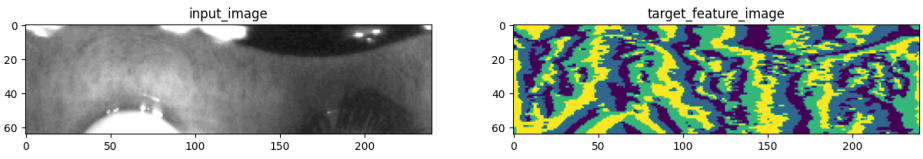


Figure 5.6: Sampling of a normalized NIR iris image and its target feature image.

The experiment demonstrates that Dice loss (1 - DICE coefficient) and IoU loss (1-Jaccard index) is more effective in training for this domain than Categorical Cross-Entropy loss. In addition, a batch size of 4 is found to be the best manner to divide the training

into batches and maximize the performance. The early stopping criteria with the patience of 10 interrupts the training at 47 revealing that the model that performs better on the validation set is at 37. Each epoch runs for 94 seconds on average on GPU P100.

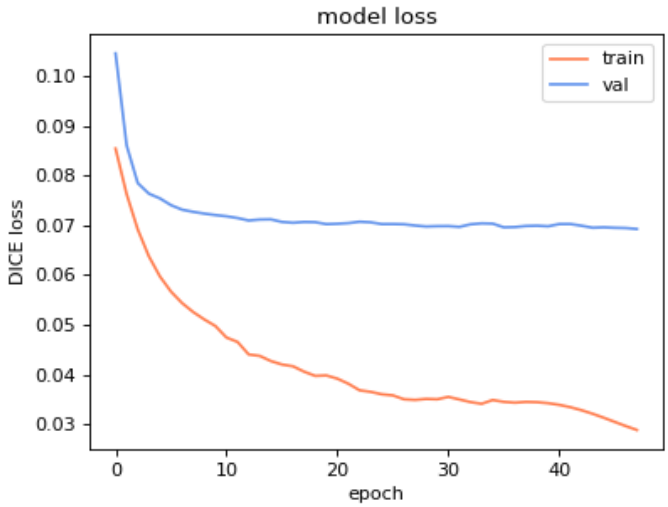


Figure 5.7: Selected model at epoch 37/47; metrics on validation set: Dice coefficient : 0.93, Jaccard index (IoU) : 0.97, Precision : 0.93, Recall : 0.93, F1-score : 0.93.

Figure 5.7 shows that the model does not suffer from overfitting even though it continues to improve on the training set. At the same time, it stabilizes and remains constant on the validation set. The Jaccard index and the Dice coefficient along with Accuracy and F1-score are the metrics used to evaluate the models during selection and final evaluation. To properly evaluate the performance and ensure that the reported metrics are representative of its generalization capability, the selected model is tested on the test set. Table 5.4 proves that the model is robust and reliable in generalizing on unseen data, learning underlying data distribution rather than memorizing specific examples since the error is close to that obtained on the validation set.

	Dice	IoU	Accuracy	F1-score
fe_model	0.93	0.87	0.96	0.93

Table 5.4: Evaluation of the selected model on the test set.

Image transcoding The dataset comprises 4, 500 samples for the training set, 900 samples for the validation set, and 870 for the test set, where each sample is a pair of 'VIS,

NIR' normalized iris images. During the pre-processing, the training set is augmented to 18,000 samples. The Figure below shows a pair picked from the training set after the pre-processing step.

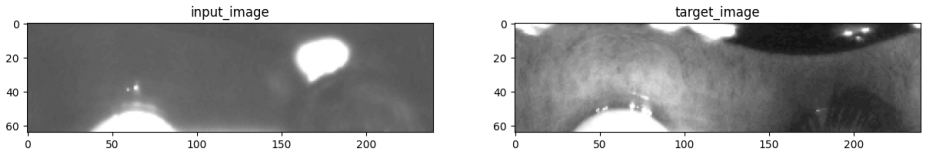


Figure 5.8: Sampling of a pair of "VIS, NIR" iris images.

As in the previous task, architectures based on U-Net-based (UNB) and Pix2pix-based (PPB) are illustrated to address the problem (see section 4.3.2): the models instantiated by these architectures share several design choices, although PPB models need to configure more hyperparameters due to their architectural complexity. The PPB models utilize the cross-entropy loss function to measure the predicted probability distribution of the model with the true probability distribution. This is expressed as $\mathcal{L}_{real\_dist}(D, F)$ and $\mathcal{L}_{fake\_dist}(G, D, F)$ in the formula 4.5. Furthermore, the L1 loss is employed as $\mathcal{L}_{pw}(G)$, which encourages less blurring in results by forcing low-frequency correctness. The parameter $\lambda_P = 100$ is chosen as it is more effective in balancing the two terms of the loss function (see formula 4.5).

	SSIM	PSNR
UNB_λ_F=0_model$_{t2}$	0.29 ± 0.01	14.86 ± 3.16
UNB_LF=Dice_λ_F=0.5_model$_{t2}$	0.29 ± 0.01	14.89 ± 3.18
UNB_LF=Dice_λ_F=1_model$_{t2}$	**0.29 ± 0.01**	**14.94 ± 3.14**
UNB_LF=IoU_λ_F=0.5_model$_{t2}$	0.29 ± 0.01	14.89 ± 3.10
UNB_LF=IoU_λ_F=1_model$_{t2}$	0.29 ± 0.01	14.84 ± 3.07
PPB_λ_F=0_model$_{t2}$	0.25 ± 0.01	14.57 ± 3.01
PPB_LF=Dice_λ_F=0.5_model$_{t2}$	0.26 ± 0.01	14.57 ± 2.87
PPB_LF=Dice_λ_F=1_model$_{t2}$	**0.27 ± 0.01**	**14.72 ± 2.87**
PPB_LF=IoU_λ_F=0.5_model$_{t2}$	0.26 ± 0.01	14.62 ± 2.89
PPB_LF=IoU_λ_F=1_model$_{t2}$	0.25 ± 0.01	14.51 ± 2.86

Table 5.5: Evaluation on the validation set focused on the I2I translation task.

Both architectures share some common hyper-parameters for model selection, which include $\mathcal{L}_{fb}(G, F)$ to measure the distance between extracted features from synthetic images and target images, λ_F which weighs this contribution in the loss function, and the batch size. The optimal batch size for both architectures is found to be 4 while λ_F and $\mathcal{L}_{fb}(G, F)$ (referred to as *LF* in Tables 5.5 and 5.6) are involved in the model selection. An early stopping regularization technique is enabled with patience of 10 for all the models.

	Dec	EER
UNB_λ_F=0_model$_{t2}$	1.6150	0.2095
UNB_LF=Dice_λ_F=0.5_model$_{t2}$	**1.6720**	**0.1171**
UNB_LF=Dice_λ_F=1_model$_{t2}$	1.6294	0.2669
UNB_LF=IoU_λ_F=0.5_model$_{t2}$	1.6513	0.1240
UNB_LF=IoU_λ_F=1_model$_{t2}$	1.6135	0.2490
PPB_λ_F=0_model$_{t2}$	1.5266	0.1568
PPB_LF=Dice_λ_F=0.5_model$_{t2}$	1.6135	0.2490
PPB_LF=Dice_λ_F=1_model$_{t2}$	**1.5716**	**0.1233**
PPB_LF=IoU_λ_F=0.5_model$_{t2}$	1.4979	0.2820
PPB_LF=IoU_λ_F=1_model$_{t2}$	1.5403	0.1390

Table 5.6: Comparison of iris recognition rates using synthetic iris images produced by the selected models, using the validation set.

As delineated in Section 5.3, the evaluation of models extends across two critical dimensions: assessing both the realism and similarity between synthetic and target images, as well as evaluating iris recognition performance. The Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) are the main indicators used to measure the quality of the transformation (see Table 5.5); these metrics serve as instrumental measures in quantifying the fidelity of the transformation process. In parallel, the evaluation extends to the domain of iris recognition, where metrics such as the Decidability (Dec) and the Equal Error Rate (EER) take a prominent role. These metrics provide valuable insights into the extent to which high-quality synthetic images contribute to improved iris recognition results, as shown in Table 5.6. This two-pronged evaluation strategy provides

a comprehensive understanding of model performance in both transformation quality and recognition effectiveness.

The process of model selection proves intricate, with no singular model demonstrating unequivocal superiority in translating normalized VIS iris images into the Near-Infrared (NIR) domain while optimizing the iris recognition rate. As illustrated in Table 5.5, the UNB models show marginally higher structural similarity index (SSIM) and peak signal-to-noise ratio (PSNR) than the PPB models, even if with minor distinctions. This little disparity is further emphasized in Table 5.6, where the UNB models show slightly higher Dec than the PPB models, despite both architectures producing EERs between 0.1 and 0.3 which is the main metric to be considered. The influence of hyper-parameters in discerning dissimilarity between synthetic and original images regarding feature image extraction is a key consideration. The results indicate that both IoU (Intersection over Union) and Dice loss produce comparable results. Figure 5.10, Figure 5.11, and Figure 5.12 showcase the inference results of the models on selected examples from the validation set. Both predictions struggle to represent iris tissue in the infrared domain partly because the iris representation in the visible domain lacks sharpness and is blurred, so in this regard, the PPB model seems to be more effective at synthesizing clearer and more defined/pronounced textures reproducing internal structures of the iris.

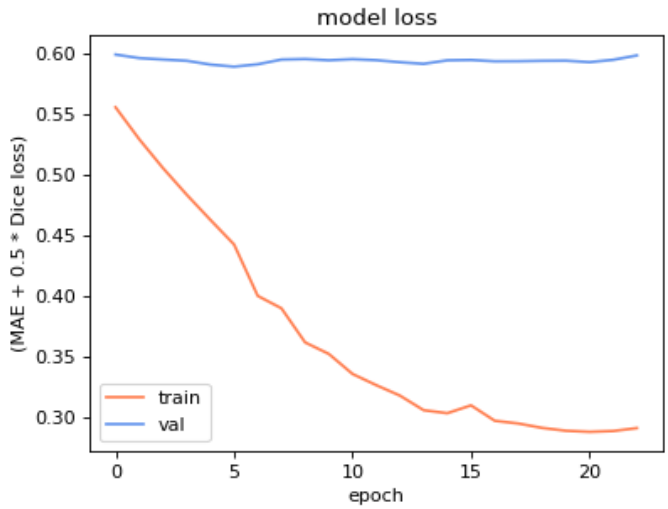


Figure 5.9: The *UNB_LF=Dice_λ_F=0.5_model*$_{t2}$ at epoch 12/22; metrics on validation set: MAE : 0.2679, Dice Loss : 0.62.

Since all models seem to have the same behavior, *UNB_LF=Dice_λ_F=0.5_model*$_{t2}$ is chosen as a representative to help understand the training. The training lasts for 22 epochs with an average run time of 602 seconds per epoch on GPU P100. The 5.9 shows that training is effective in decreasing the error on the training set, thus learning the

training data very well, but does not generalize well to new unseen data. The large gap between the training and validation curves suggests overfitting; this is probably because of the complexity of the model and insufficient data that make it difficult for the model to generalize well. The plateaued validation curve justifies why models with $\lambda_F = 0$ (which neglect the comparison between feature images and focus only on the iris representation in the loss function) and $0 < \lambda_F <= 1$ are close to each other. As mentioned above, both architectures fit the problem in the same way also exposed through the graph of the training of *UNB_LF=Dice_λ_F=0.5_model*$_{t2}$. Therefore, during the model selection phase, the best models of both architectures are selected and subjected to the testing phase. Although the results are all very close, the Dice loss as $\mathcal{L}_{f,b}(G, F)$ in the loss function obtains higher Dec in both architectures, while $\lambda_F = 0.5$ seems to be more effective for the UNB architecture and $\lambda_F = 1$ for PPB obtaining lower EERs.

	SSIM	PSNR
UNB_LF=Dice_λ_F=0.5_model$_{t2}$	**0.24 ± 0.01**	**13.68 ± 5.16**
PPB_LF=Dice_λ_F=1_model$_{t2}$	0.22 ± 4.41	13.46 ± 4.41

Table 5.7: Evaluation on the test set focused on the I2I translation task.

	Dec	EER
normalized VIS iris images	1.2880	0.3399
normalized NIR iris images	1.4681	0.0394
UNB_LF=Dice_λ_F=0.5_model$_{t2}$	1.2410	0.3772
PPB_LF=Dice_λ_F=1_model$_{t2}$	**1.1671**	**0.1203**

Table 5.8: Comparison of iris recognition rates using synthetic iris images produced by the selected models and the original iris images from the test set.

The same metrics used during the model selection are used to evaluate the performance of the models. Although SSIM and PSNR are already low, they are slightly lower on the test set. This emphasizes even more how the training phase is not excelling. To evaluate the iris recognition performance, the results obtained are also compared with the recognition results obtained using the original images. '*normalized VIS iris images*' represents the test set of original normalized iris images acquired under visible light while '*normalized NIR iris images*' acquired near-infrared light. The last two entries, on the other hand, are

devoted to the selected deep-learning models that take the test set in visible light and translate it into the near-infrared spectrum. The UNB and PPB models both show lower performance in Dec compared to '*normalized VIS iris images*' appearing not to increase the quality of the source images enough; however, the PPB model obtains a significantly lower error on EER consistent with that obtained on the validation set (both are about 0.12), showing a real performance improvement in iris recognition.

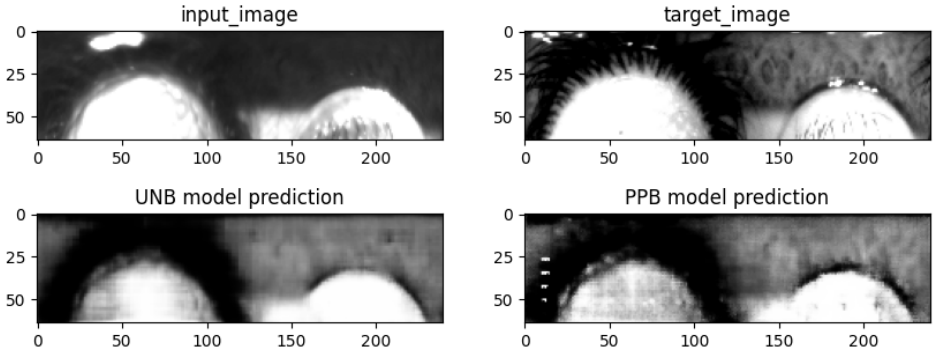


Figure 5.10: Predictions of *UNB_LF=Dice_λ_F=0.5_model*$_{t2}$ and *PPB_LF=Dice_λ_F=1_model*$_{t2}$ on subject 151, left eye, first acquisition.

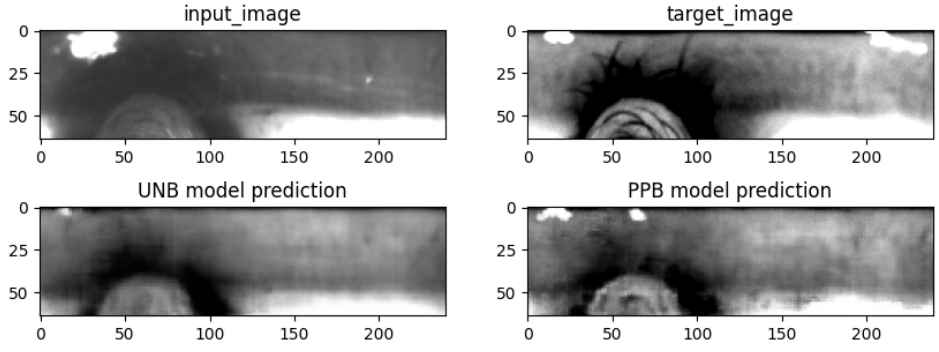


Figure 5.11: Predictions of *UNB_LF=Dice_λ_F=0.5_model*$_{t2}$ and *PPB_LF=Dice_λ_F=1_model*$_{t2}$ on subject 153, right eye, first acquisition.

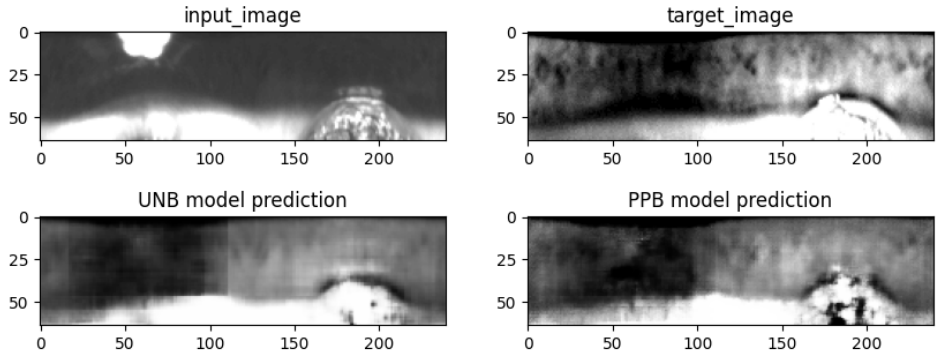


Figure 5.12: Predictions of *UNB_LF=Dice_λ_F=0.5_model*$_{t2}$ and *PPB_LF=Dice_λ_F=1_model*$_{t2}$ on subject 172, left eye, forth acquisition.

5.4.3 Models for extracting more discriminating features

For this task, named $t3$, the dataset comprises only normalized VIS iris images adhering to the same division into sets expressed in Section 5.1.1. The division encompasses 4, 500 samples designated for the training set, 900 samples for the validation set, and 870 for the test set. Each normalized VIS iris image is associated with a feature image which is the result of the algorithm described in Section 4.2 that takes as input the normalized iris image, but in the NIR spectrum. These feature images serve as ground truth during the training process. To provide a visual representation, the figure below displays an image randomly selected from the training set after undergoing the pre-processing step, along with its associated feature image:

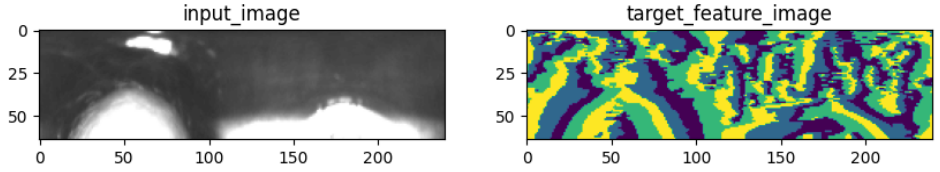


Figure 5.13: Sampling of a normalized VIS iris image and its NIR feature image.

The task is very similar to these presented in Section 4.3.1; in that case, the architecture shapes a feature extractor that replaces the algorithm described in Section 4.2 learning a mapping between normalized NIR iris images and their feature images. In this case, the architecture proposed in Section 4.4 aims to find correspondences between normalized VIS iris images and the NIR feature images to directly obtain high-quality features.

U-Net-based (UNB) and Pix2pix-based (PPB) architectures are proposed to tackle the problem, as outlined in Section 4.4. Both architectures share certain design choices, PPB models require more extensive configuration of hyper-parameters due to their increased

	Accuracy	F1-score
UNB_LF=Dice_model$_{t3}$	**0.69**	**0.38**
UNB_LF=IoU_model$_{t3}$	0.69	0.38
UNB_LF=Scce_model$_{t3}$	0.69	0.38
PPB_LF=Dice_model$_{t3}$	**0.69**	**0.38**
PPB_LF=IoU_model$_{t3}$	0.69	0.38
PPB_LF=Scce_model$_{t3}$	0.69	0.38

Table 5.9: Evaluation on the validation set focused on the per-pixel classification task.

architectural complexity. Specifically, PPB models employ the cross-entropy loss function, denoted as $\mathcal{L}_{real\_dist}(D)$ and $\mathcal{L}_{fake\_dist}(G, D)$ in the formula 4.6, to evaluate the alignment between the predicted probability distribution generated by the model and the true probability distribution. Additionally, λ_P is set to 5 offering an effective balance between the two terms in the loss function (see the formula 4.7). Both architectures share common hyper-parameters for model selection, including $\mathcal{L}_{mc}(G)$ to measure the distance between extracted features from synthetic images and target images and the batch size. The optimal batch size for both architectures is determined to be 4, while $\mathcal{L}_{mc}(G)$ loss plays a role in the model selection. An early stopping regularization technique is applied with a patience of 10 for all models.

	Dec	EER
UNB_LF=Dice_model$_{t3}$	**1.9019**	**0.1078**
UNB_LF=IoU_model$_{t3}$	1.9108	0.1348
UNB_LF=Scce_model$_{t3}$	1.8846	0.1417
PPB_LF=Dice_model$_{t3}$	**1.8921**	**0.1348**
PPB_LF=IoU_model$_{t3}$	1.9288	0.2394
PPB_LF=Scce_model$_{t3}$	1.3790	0.1350

Table 5.10: Evaluation on the validation set comparing iris recognition performance using synthetic feature images.

As indicated in the section 5.3, the evaluation of the models comes in two different

directions: a first evaluation includes metrics such as Accuracy to measure the overall correctness of the model by calculating the ratio of correctly predicted pixels to the total number of pixels, and F1-score to provide further insights by considering both precision and recall. In parallel, metrics such as Decidability (Dec) and Equal Error Rate (EER) help to understand whether the model contributes to synthesizing more discriminative image features.

The process of selecting the optimal model proves to be complex, with no single model demonstrating clear superiority in synthesizing NIR-like feature images. As shown in Table 5.9, UNB and PPB models are completely stuck on the same Accuracy and F1-score. Table 5.10 confirms equality in results in terms of Dec and EER. The impact of the loss function on discerning dissimilarity between synthetic and target feature images is a crucial consideration. Results indicate that Intersection over Union (IoU), Dice loss, and sparse categorical cross entropy (SCCE) yield comparable outcomes, even if the Dice loss seems to obtain slightly higher results in terms of EERs in both architectures. Figures 5.15, 5.16, and 5.17 display the inference results of the models on selected examples from the validation set.

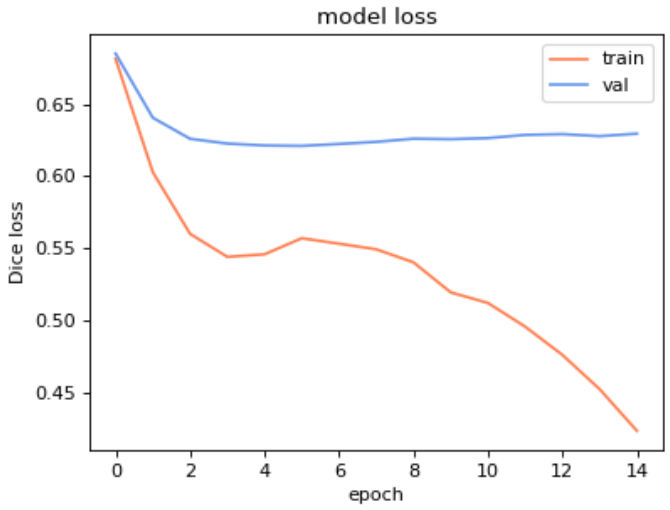


Figure 5.14: The *PPB_LF=Dice_model*$_{t3}$ at epoch 4/14; metrics on validation set: Dice Loss : 0.62, Accuracy : 0.69 , F1-score : 0.28.

Since all models exhibit similar behavior, *PPB_LF=Dice_model*$_{t3}$ is chosen as a representative to aid in understanding the training process. The training spans 14 epochs with an average runtime of 203 seconds per epoch on GPU P100. Figure 5.14 indicates that the training effectively reduces the error on the training set, demonstrating a thorough understanding of the training data. However, the model struggles to generalize well to new, unseen data, as evidenced by the significant gap between the training and validation

curves, suggesting overfitting. Both architectures approach the problem similarly, as depicted in the graph. Hence, during the model selection, the top-performing models from both architectures are chosen for further evaluation in the testing phase.

	Accuracy	F1-score
UNB_LF=Dice_model$_{t3}$	0.68	0.35
PPB_LF=Dice_model$_{t3}$	**0.68**	**0.35**

Table 5.11: Evaluation on the test set focused on the pixel classification task.

	Dec	EER
VIS feature images	1.2880	0.3399
NIR feature images	1.4681	0.0394
UNB_LF=Dice_model$_{t3}$	1.3746	0.1123
PPB_LF=Dice_model$_{t3}$	**1.3638**	**0.1009**

Table 5.12: Evaluation of iris recognition performance on feature images from normalized VIS and NIR iris images, comparing them with feature images generated by selected models, using the test set.

The assessment of model performance employs the same metrics used during the model selection process. The Accuracy and F1-score are found to be similar to the results obtained on the validation set, albeit at a relatively low level. This reveals that the models are restricted in their ability to directly approximate NIR feature images, as evidenced by the learning in Figure 5.14. To evaluate recognition performance, the obtained results are compared with recognition results derived from feature images extracted from the original images. The test set comprises normalized iris images acquired in visible light (VIS) and near-infrared (NIR), where the '*VIS feature images*' and '*NIR feature images*' represent their respective feature images. The final two entries in Table 5.12 focus on selected deep learning models tasked with extracting NIR-like feature images. Both UNB and PPC models exhibit higher performance in terms of Dec and EER compared to the '*VIS feature images*', which suggests that the models do enhance the quality of the feature images even though the Accuracy and F1-score indicate that the NIR-like representations still fall short of the original ones.

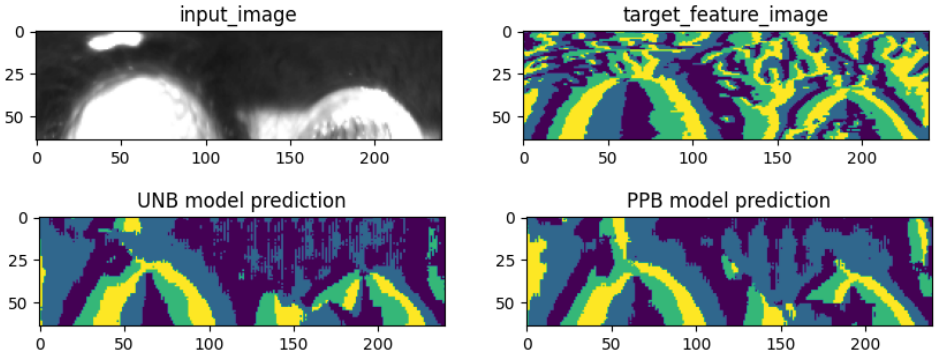


Figure 5.15: Predictions of *UNB_LF=Dice_model*$_{t3}$ and *PPB_LF=Dice_model*$_{t3}$ on subject 151, left eye, first acquisition.

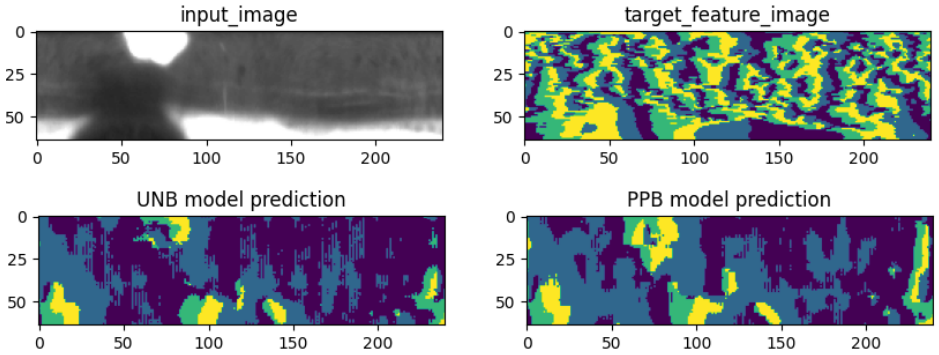


Figure 5.16: Predictions of *UNB_LF=Dice_model*$_{t3}$ and *PPB_LF=Dice_model*$_{t3}$ on subject 153, right eye, first acquisition.

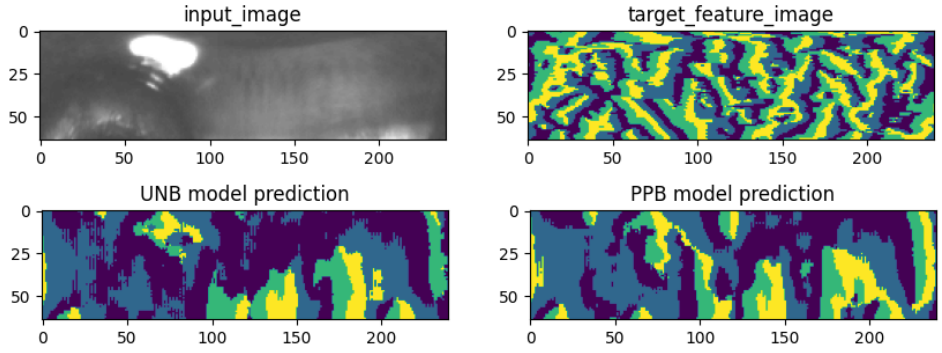


Figure 5.17: Predictions of *UNB_LF=Dice_model*$_{t3}$ and *PPB_LF=Dice_model*$_{t3}$ on subject 167, left eye, forth acquisition.

5.5 Combined evaluation of Models

In the previous section, each model is tested for its task, providing a solid evaluation of their suitability for the given problem. In this section, the models are tested together to

evaluate their overall effectiveness in terms of iris recognition. The transcoding models, which respectively apply visible to near-infrared domain transcoding to improve iris segmentation (see Section 5.4.1) and feature extraction (see Section 5.4.2), are integrated into the pipeline of an iris recognition system (see 5.18). To obtain a complete understanding of the effectiveness of the approach, the test set, which contains the iris images of 29 subjects, where each subject has 4 acquisitions per eye, is used to evaluate the iris recognition performance. The system employs the IS_{IS} method for iris segmentation, Daugman's classical rubber-sheet method for the normalization step, while feature extraction and matching follow the procedures described in Section 5.3. The system undergoes three separate executions: first, using all 'VIS iris images', then employing all 'NIR iris images', and finally, reprocessing the VIS iris images. In the latter case, "transcoding model $t1$" and "transcoding model $t2$" are enabled (see Figure 5.18), and Table 5.13 reports different entries for the different combinations of models. Each execution produces Dec and EER as metrics to evaluate the system. For task $t1$, a single model is selected, while for task $t2$, two models are chosen, each based on different architectural choices that yield similar results during separate evaluation (see Table 5.6).

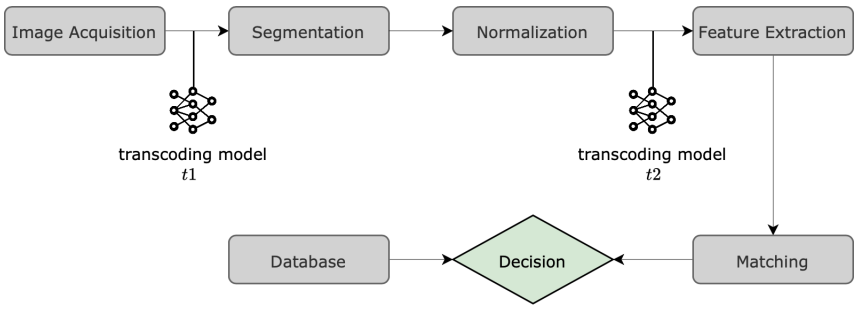


Figure 5.18: Transcoding models inside the iris recognition system.

When the image transcoding steps are enabled, a design choice must be made after the normalization step. Specifically, it involves choosing whether to treat the normalized VIS iris images as gray-scale or RGB images. The input images have three channels as they were acquired under visible light, but the models for task $t2$ are trained with normalized VIS iris images in gray-scale representation. Following the standard process, a strategy would transform the normalized VIS iris images to gray-scale before feeding them to the $t2$ models. Alternatively, the images could be left in three channels. In this case, the image transcoding would be applied to each channel separately, resulting in the extraction of feature images with three channels. The table 5.13 illustrates the selection of the optimal pair of models by opting for '<UNB_model_{t1}, $PPB\_LF{=}Dice\_\lambda_F{=}1\_model_{t2}$>'. Despite having a lower Dec, it exhibits a superior EER that is very close to the results obtained during the individual evaluation step of the $model_{t2}$. This consistency and generalization ability make this pair the preferred choice. Additionally, the computational

effort associated with performing 'image transcoding 2' on each image channel does not yield significant benefits in terms of results. This observation underscores the convenience and cost-effectiveness of dealing with gray-scale images.

		Dec	EER
VIS iris images		0.9752	0.5246
NIR iris images		1.5371	0.0993
UNB_model_{t1} + $UNB\_LF{=}Dice\_\lambda_F{=}0.5\_model_{t1}$	g	0.9259	0.3389
	rgb	0.9337	0.5349
UNB_model$_{t1}$ + ***PPB_LF=Dice***$\_\lambda_F$***=1_model***$_{t2}$	g	**0.8728**	**0.1462**
	rgb	0.8000	0.1946

Table 5.13: Evaluation on the test set to compare the performance of the iris recognition system with transcoding models.

In conclusion, the final phase of the thesis project aims to streamline the architecture introduced in Section 4.3 by attempting to directly construct a novel CNN-based feature extractor leveraging the inherent relationship between VIS and NIR spectra (see Section 4.4). However, this imposes a constraint on iris recognition systems (see Figure 5.19). This leaves the segmentation system as the primary aspect where implementation choices remain flexible and open.

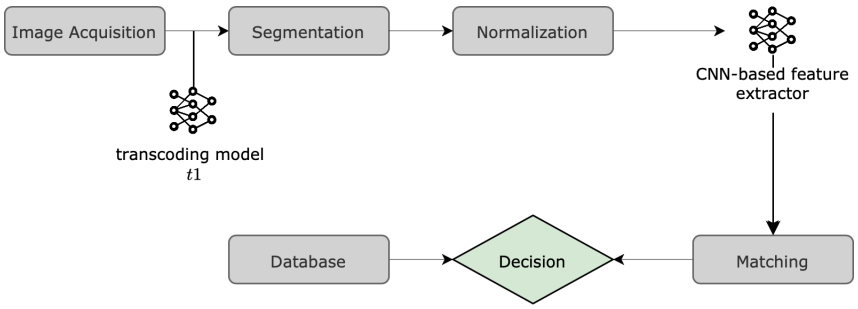


Figure 5.19: The iris recognition system constrained to the use of CNN-based feature extraction model.

Table 5.14 illustrates the results. The initial two entries are computed using the same procedure described for the previous evaluation. The architecture in Figure 5.19 is used

only for executions involving deep-learning models, where 'transcoding model $t1$' is incorporated, and the feature extraction phase is replaced by $model_{t3}$. Once again, feature extraction is carried out for both gray-scale and RGB image interpretations. Each run yields Dec and EER values. For task $t1$, a single model is chosen, while for task $t3$, two models are selected, each based on distinct architectural choices that result in similar performance during individual evaluations (see Table 5.12). Table 5.14 showcases the selection of the model pair '<UNB_model_{t1}, $PPB\_LF{=}Dice\_model_{t3}$>'. This pair exhibits superior EER, closely resembling the results obtained from evaluating the single model $t3$, thus affirming consistency and generalization. The computational efficiency of processing the normalized VIS iris images as RGB images during feature extraction is very effective in this scenario and leads to a significant reduction of EER. In summary, the model pairs chosen from Table 5.13 and Table 5.14 effectively decrease the Equal Error Rate (EER) when the recognition system processes VIS iris images from the test set as input, approaching the performance achieved when considering images in the NIR spectrum. The streamlined architecture with the feature extractor ($model_{t3}$) outperforms the transcoding approach ($model_{t2}$), achieving even better results. However, it imposes limitations on the recognition system by prescribing a specific feature extraction strategy.

		Dec	EER
VIS iris images		0.9752	0.5246
NIR iris images		1.5371	0.0993
UNB_model_{t1} + $UNB\_LF{=}Dice\_model_{t3}$	g	1.0162	0.4700
	rgb	1.0322	0.2218
UNB_model$_{t1}$ + ***PPB_LF=Dice_model***$_{t3}$	g	1.0033	0.3989
	rgb	**1.0151**	**0.1004**

Table 5.14: Evaluation on the test set to compare the performance of the iris recognition system with CNN-based feature extractor.

–6–
Conclusion and future developments

The thesis introduced CNN-based transcoding models tailored to improve the performance of iris recognition systems that operate in the visible (VIS) spectrum. A primary objective, denoted as task $t1$ involved in enhancing image quality by translating VIS iris images in the near-infrared (NIR) spectrum. This reconstruction aimed to mitigate noise factors that often interfere with iris segmentation when using visible light. After that, the focus shifted to normalized (segmented) VIS iris images, increasing the detail of iris textures by projecting them into the NIR spectrum. This reconstruction aimed to extract more detailed information and discriminative features, defining task $t2$. A novel loss function was implemented, measuring the reconstruction error of the normalized NIR-like iris images by evaluating the feature similarity extracted by a Daugman-based algorithm. The final phase of the research, denoted as task $t3$ proposed a novel feature extraction method inspired by a Daugman-based approach. The proposed method aimed to generate high-quality feature images from normalized VIS iris images, similar to those extracted from their representations in the NIR spectrum, by exploiting the VIS-NIR relationship as task $t1$ and task $t2$ did for image transcoding. State-of-the-art neural network architectures in Image-to-Image (I2I) translation inspired the design of neural network models: U-Net and Pix2pix. The evaluation of tasks was conducted independently before integrating them into the pipeline of a traditional iris recognition system. Key architectural choices included utilizing the IS_{IS} method for iris segmentation and a Daugman-based algorithm for feature extraction.

Experiments conducted on task $t1$ revealed that the U-Net-based (UNB) model outperformed the Pix2pix-based (PPB) model in translating iris images into the NIR spectrum, being very effective in reducing glare and reflections, controlling different illumination conditions, and highlighting the differentiation between the iris and surrounding structures. The UNB model obtained an SSIM of 0.87 ± 0.01 and a PSNR of 25.03 ± 13.05. In addition, the UNB model was tested on a cross-dataset using the MICHE datasets and the IS_{IS} method to assess the improvement in iris segmentation accuracy after image

transcoding. The approach yielded limited effectiveness, primarily due to the inherent challenge of surpassing the remarkable results achieved by the IS_{IS} method on the original images, which demonstrated exceptional accuracy with a Dice coefficient of 0.96 ± 0.05 and an F1 score of 0.96 ± 0.05.

Experiments conducted on task $t2$ revealed that the UNB and PPB models suffered from overfitting and thus had inadequate training. This issue was primarily derived from limited available data and the subsequent complexity of the models. The PPB model obtained an SSIM of 0.24 ± 0.01 and a PSNR of 13.68 ± 5.16. Despite the poor qualitative results on image reconstruction, the model still performed well in terms of iris recognition. The normalized VIS iris images processed by the model obtained an EER of 0.12, showcasing a significant increase in performance obtained on the original images, which was 0.34.

In the conclusive experiment, the models derived from tasks $t1$ and $t2$ were integrated into the iris recognition pipeline. The baseline iris recognition system achieved a Decidability (Dec) of 0.98 and an equal error rate (EER) of 0.52. Enabling the two proposed models for image transcoding within the workflow, the system achieved a Dec of 0.87 and an EER of 0.15. This reduction in EER highlighted the substantial efficacy of transcoding models in minimizing recognition errors, bringing the system's performance closer to the error levels obtained by repeating the experiment with near-infrared (NIR) iris images, where the EER was 0.10.

Experiments conducted for task $t3$ revealed that both UNB and PPB encountered challenges related to overfitting, failing to further improve the error on the validation set. Despite the limitations in generating synthetic feature images close to real NIR representations of the best PPB model with an Accuracy of 0.68 and an F1-score of 0.35, there was a noticeable improvement in the quality of the extracted features, as evidenced by the performance in recognition tasks. The PPB model obtained a Dec of 1.36 and EER of 0.10, while Daugman-based feature extraction on normalized VIS iris images resulted in a Dec of 1.29 and an EER of 0.34.

In the conclusive experiment, the model derived from tasks $t1$ and $t3$ formed a new iris recognition system. The baseline recognition system obtained a Dec of 0.98 and an EER of 0.52. By enabling the image-processing model of task $t1$ and replacing the CNN-based feature extractor provided by task $t3$, the system reached a Dec of 1.02 and an EER of 0.10, improved considerably.

Future advancements in this research encompass several crucial aspects. Foremost, a database of iris image pairs acquired under both visible and near-infrared light, with increased size and variability. This enhancement is particularly relevant for task $t1$ facilitating the training of more robust models capable of handling additional noise factors such as mirror reflections, off-axis irises, blurring, or artifacts. The current dataset used for model training is limited in addressing these potential challenges comprehensively.

Furthermore, scaling up the dataset and leveraging additional computational resources, would contribute to the optimal training of models produced by the tasks $t2$ and $t3$, which have already demonstrated effectiveness. In particular, for task $t2$, exploring the integration of a non-Daugman-based feature extractor into the model loss function could provide valuable insights, even though this might renounce some theoretical guarantees regarding the approach's feasibility. Additionally, future investigations could involve changing the recognition system used for evaluating models and integrating them into different systems, especially by trying different methods to extract the features that might prove most effective on the synthetic images.

Bibliography

[1] Jain et al. “Biometrics personal identification in networked society”. In: vol. 479. Springer Science & Business Media, 1999, pp. 4–16.

[2] Yin Y. et al. “Deep Learning for Iris Recognition: A Review”. In: *arXiv:2303.08514* (2021).

[3] Bobeldyk D. and Ross A. “Predicting eye color from near infrared iris images”. In: *International Conference on Biometrics (ICB)* (2018).

[4] “Deep gan-based cross-spectral cross-resolution iris recognition”. In: *Transactions on Biometrics, Behavior, and Identity Science* 3.4 (2021), pp. 443–463.

[5] Pang Y., Lin J., Qin T., and Chen Z. “Image-to-image translation: Methods and applications”. In: *IEEE Transactions on Multimedia* 24 (2021), pp. 3859–3881.

[6] Jain A., Bolle R., and Pankanti S. *Biometrics: personal identification in networked society*. Vol. 479. Springer Science & Business Media, 1999.

[7] Leonard F. and Aran S. “Iris recognition system”. In: *Patent, US4641349 A* (1987).

[8] Nguyen K. et al. “Long range iris recognition: A survey”. In: *Pattern Recognition* 72 (2017), pp. 123–143.

[9] Bowyer K. “The results of the NICE. II iris biometrics competition”. In: *Pattern Recognition Letters* 33.8 (2012), pp. 965–969.

[10] De Marsico M., Nappi M., Riccio D., and Wechsler H. “Mobile iris challenge evaluation (MICHE)-I, biometric iris dataset and protocols”. In: *Pattern Recognition Letters* 57 (2015), pp. 17–23.

[11] Malgheet J., Manshor N., Affendey L., and Abdul Halin A. “Iris recognition development techniques: a comprehensive review”. In: *Complexity* 2021 (2021), pp. 1–32.

[12] Wei Z., Tan T., Sun Z., and Cui J. “Robust and fast assessment of iris image quality”. In: *Advances in Biometrics: International Conference*. Springer. 2005, pp. 464–471.

[13] Kalka N., Zuo J., Schmid N., and Cukic B. “Image quality assessment for iris biometric”. In: *Biometric technology for human identification III*. Vol. 6202. SPIE. 2006, pp. 124–134.

[14] Wildes R. “Iris recognition: an emerging biometric technology”. In: *Proceedings of the IEEE* 85.9 (1997), pp. 1348–1363.

[15] Liu Y., Yuan S., Zhu X., and Cui Q. “A practical iris acquisition system and a fast edges locating algorithm in iris recognition”. In: *Instrumentation and measurement technology conference proceedings*. Vol. 1. IEEE. 2003, pp. 166–169.

[16] Huang Y., Luo S., and Chen E. “An efficient iris recognition system”. In: *Proceedings. International Conference on Machine Learning and Cybernetics*. Vol. 1. IEEE. 2002, pp. 450–454.

[17] Liu X., Bowyer K., and Flynn P. “Experiments with an improved iris segmentation algorithm”. In: *Automatic Identification Advanced Technologies (AutoID’05)*. IEEE. 2005, pp. 118–123.

[18] Lili P. and Mei X. “The algorithm of iris image preprocessing”. In: *Workshop on Automatic Identification Advanced Technologies (AutoID’05)*. IEEE. 2005, pp. 134–138.

[19] Feng X., Fang C., Ding X., and Wu Y. “Iris localization with dual coarse-to-fine strategy”. In: *18th International Conference on Pattern Recognition (ICPR’06)*. Vol. 4. IEEE. 2006, pp. 553–556.

[20] Bowyer K., Hollingsworth K., and Flynn P. “Image understanding for iris biometrics: A survey”. In: *Computer vision and image understanding* 110.2 (2008), pp. 281–307.

[21] Kong W. and Zhang D.avid. “Detecting eyelash and reflection for accurate iris segmentation”. In: *International journal of pattern recognition and artificial intelligence* 17.06 (2003), pp. 1025–1034.

[22] Boles W. and Boashash B. “A human identification technique using images of the iris and wavelet transform”. In: *transactions on signal processing* 46.4 (1998), pp. 1185–1188.

[23] Lian S. et al. “Attention guided U-Net for accurate iris segmentation”. In: *Journal of Visual Communication and Image Representation* 56 (2018), pp. 296–304.

[24] Zhang W. et al. “A robust iris segmentation scheme based on improved U-net”. In: *IEEE Access* 7 (2019), pp. 85082–85089.

[25] Wu X. and Zhao L. “Study on iris segmentation algorithm based on dense U-Net”. In: *IEEE Access* 7 (2019), pp. 123959–123968.

[26] Wang C. et al. “Towards complete and accurate iris segmentation using deep multi-task attention network for non-cooperative iris recognition”. In: *IEEE Transactions on information forensics and security* 15 (2020), pp. 2944–2959.

[27] Patil S., Jha R., and Nigam A. "IpSegNet: deep convolutional neural network based segmentation framework for iris and pupil". In: *13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE. 2017, pp. 184–191.

[28] Rot P., Ž. Emeršič, Struc V., and Peer P. "Deep multi-class eye segmentation for ocular biometrics". In: *international work conference on bioinspired intelligence (IWOBI)*. IEEE. 2018, pp. 1–8.

[29] Korobkin M. et al. "Iris segmentation in challenging conditions". In: *Pattern Recognition and Image Analysis* 28 (2018), pp. 652–657.

[30] Arsalan M. et al. "IrisDenseNet: Robust iris segmentation using densely connected fully convolutional networks in the images by visible light and near-infrared light camera sensors". In: *Sensors* 18.5 (2018), p. 1501.

[31] Arsalan M. et al. "FRED-Net: Fully residual encoder–decoder network for accurate iris segmentation". In: *Expert Systems with Applications* 122 (2019), pp. 217–241.

[32] Li Y., Huang P., Juan Y., et al. "An efficient and robust iris segmentation algorithm using deep learning". In: *Mobile Information Systems* 2019 (2019).

[33] Zhao Z. and Kumar A. "A deep learning based unified framework to detect, segment and recognize irises using spatially corresponding features". In: *Pattern Recognition* 93 (2019), pp. 546–557.

[34] Liu N. et al. "Accurate iris segmentation in non-cooperative environments using fully convolutional networks". In: *International Conference on Biometrics (ICB)*. IEEE. 2016, pp. 1–8.

[35] He Y. et al. "Visible spectral Iris segmentation via deep convolutional network". In: *Biometric Recognition: 12th Chinese Conference, CCBR 2017, Shenzhen, China, October 28-29, 2017, Proceedings 12*. Springer. 2017, pp. 428–435.

[36] Lozej J., Š. D., Štruc V., and Peer P. "Influence of segmentation on deep iris recognition performance". In: *7th International Workshop on Biometrics and Forensics (IWBF)*. IEEE. 2019, pp. 1–6.

[37] Abbasi M. "Improving identification performance in iris recognition systems through combined feature extraction based on binary genetics". In: *SN Applied Sciences* 1.7 (2019), p. 730.

[38] Shamsi M. and Rasouli A. "An innovative trapezium normalization for iris recognition systems". In: *Int Conf Comput Softw Model*. Vol. 14. 2011, pp. 130–134.

[39] Kekre HB. et al. "Iris recognition using partial coefficients by applying discrete cosine transform, haar wavelet and DCT wavelet transform". In: *International Journal of Computer Applications* 32.6 (2011), pp. 0975–8887.

[40] Jang J., Park K., Kim J., and Lee Y. “New focus assessment method for iris recognition systems”. In: *Pattern recognition letters* 29.13 (2008), pp. 1759–1767.

[41] Lowe D. “Distinctive image features from scale-invariant keypoints”. In: *International journal of computer vision* 60 (2004), pp. 91–110.

[42] Barpanda S. et al. “Iris recognition with tunable filter bank based feature”. In: *Multimedia Tools and Applications* 77.6 (2018), pp. 7637–7674.

[43] Barpanda S. et al. “Iris feature extraction through wavelet mel-frequency cepstrum coefficients”. In: *Optics & Laser Technology* 110 (2019), pp. 13–23.

[44] Krizhevsky A., Sutskever I., and Hinton G. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).

[45] Sharif Razavian A., Azizpour H., Sullivan J., and Carlsson S. “CNN features off-the-shelf: an astounding baseline for recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2014, pp. 806–813.

[46] Simonyan K. and Zisserman A. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv:1409.1556* (2014).

[47] Carvalho T. et al. “Exposing computer generated images by eye’s region classification via transfer learning of VGG19 CNN”. In: *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE. 2017, pp. 866–870.

[48] Minaee S., Abdolrashidiy A., and Wang Y. “An experimental study of deep convolutional features for iris recognition”. In: *Signal processing in medicine and biology symposium (SPMB)*. IEEE. 2016, pp. 1–6.

[49] Reddy N., Rattani A., and Derakhshani R. “Ocularnet: deep patch-based ocular biometric recognition”. In: *2018 IEEE international symposium on technologies for homeland security (HST)*. IEEE. 2018, pp. 1–6.

[50] El-Sayed M. and Abdel-Latif M. “Iris recognition approach for identity verification with DWT and multiclass SVM”. In: *PeerJ Computer Science* 8 (2022), e919.

[51] Rana H., Azam M., and Akhtar M. “Iris recognition system using PCA based on DWT”. In: *SM Journal of Biometrics & Biostatistics* 2.3 (2017), p. 1015.

[52] Daouk CH, El-Esber LA, Kammoun FD, and Al Alaoui MA. “Iris recognition”. In: *IEEE ISSPIT*. 4. 2002, p. 558.

[53] Chirchi V., Waghmare LM, and Chirchi ER. “Iris biometric recognition for person identification in security systems”. In: *International Journal of Computer Applications* 24.9 (2011), pp. 1–6.

[54] Yiming Z. and Jun W. "Research on iris recognition algorithm based on hough transform". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 439. 3. IOP Publishing. 2018, p. 032007.

[55] Hanfei L. and Congfeng J. "One Hamming Distance Deviation Matching Approach for Iris Recognition". In: *International Journal of Security and Its Applications* 8.5 (2014), pp. 277–290.

[56] Zhao Z. and Kumar A. "Towards more accurate iris recognition using deeply learned spatially corresponding features". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 3809–3818.

[57] Marra F., Poggi G., Sansone C., and Verdoliva L. "A deep learning approach for iris sensor model identification". In: *Pattern Recognition Letters* 113 (2018), pp. 46–53.

[58] Zhu J., Park T., Isola P., and Efros A. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

[59] Wang L. et al. "A state-of-the-art review on image synthesis with generative adversarial networks". In: 8 (2020), pp. 63514–63537.

[60] Wang T. et al. "High-resolution image synthesis and semantic manipulation with conditional gans". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8798–8807.

[61] Li R. et al. "Simplified unsupervised image translation for semantic segmentation adaptation". In: *Pattern Recognition* 105 (2020), p. 107343.

[62] Yuan Y. et al. "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 701–710.

[63] Graffieti G. "Style Transfer with Generative Adversarial Networks". Master thesis. University of Bologna, 2018.

[64] Xu Z. et al. "Learning from multi-domain artistic images for arbitrary style transfer". In: *arXiv:1805.09987* ().

[65] Harshvardhan GM., Gourisaria M., Pandey M., and Rautaray S. "A comprehensive survey and analysis of generative models in machine learning". In: *Computer Science Review* 38 (2020), p. 100285.

[66] Ng A. and Jordan M. "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes". In: *Advances in neural information processing systems* 14 (2001).

[67] Isola P., Zhu J., Zhou T., and Efros A. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 1125–1134.

[68] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes. stat, vol. 1050". In: (2014).

[69] Cemgil T. et al. "The autoencoding variational autoencoder". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15077–15087.

[70] Goodfellow I. et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[71] Wang K. et al. "Generative adversarial networks: introduction and outlook". In: *IEEE/CAA Journal of Automatica Sinica* 4.4 (2017), pp. 588–598.

[72] Oussidi A. and Elhassouny A. "Deep generative models: Survey". In: *International conference on intelligent systems and computer vision (ISCV)* (2018), pp. 1–8.

[73] Barnett S. "Convergence problems with generative adversarial networks (gans)". In: *arXiv:1806.11382* (2018).

[74] Mirza M. and Osindero S. "Conditional generative adversarial nets". In: *arXiv:1411.1784* (2014).

[75] Ronneberger O., Fischer P., and Brox T. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention* (2015), pp. 234–241.

[76] Liu F. and Wang L. "UNet-based model for crack detection integrating visual explanations". In: *Construction and Building Materials* 322 (2022), p. 126265.

[77] Demir U. and Unal G. "Patch-based image inpainting with generative adversarial networks". In: *arXiv:1803.07422* (2018).

[78] Alotaibi A. "Deep generative adversarial networks for image-to-image translation: A review". In: *Symmetry* 12.10 (2020), p. 1705.

[79] Nilsson J. and Akenine-Möller T. "Understanding ssim". In: *arXiv:2006.13846* (2020).

[80] Salimans T. et al. "Improved techniques for training gans". In: *Advances in neural information processing systems* 29 (2016).

[81] Nsaef A., Jaafar A., and Jassim K. "Enhancement segmentation technique for iris recognition system based on Daugman's Integro-differential operator". In: *International Symposium on Instrumentation & Measurement, Sensor Network and Automation (IMSNA)*. Vol. 1. IEEE. 2012, pp. 71–75.

[82] *The Hong Kong Polytechnic University Cross-Spectral Iris Image Database*. `https://www4.comp.polyu.edu.hk/~csajaykr/polyuiris.htm`. 2015.

[83] Reed R. and MarksII R. *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1999.

[84] Trebing K., Stanczyk T., and Mehrkanoon S. "SmaAt-UNet: Precipitation nowcasting using a small attention-UNet architecture". In: *Pattern Recognition Letters* 145 (2021), pp. 178–186.

[85] Johnson J., Alahi A., and Fei-Fei L. "Perceptual losses for real-time style transfer and super-resolution". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer. 2016, pp. 694–711.

[86] Abu-Srhan A., Abushariah M., and Al-Kadi O. "The effect of loss function on conditional generative adversarial networks". In: *Journal of King Saud University-Computer and Information Sciences* 34.9 (2022), pp. 6977–6988.

[87] Hosseini M., Araabi B., and Soltanian-Zadeh H. "Pigment melanin: Pattern for iris recognition". In: *transactions on instrumentation and measurement* 59.4 (2010), pp. 792–804.

[88] Ma L., Tan T., Wang Y., and Zhang D. "Personal identification based on iris texture analysis". In: *IEEE transactions on pattern analysis and machine intelligence* 25.12 (2003), pp. 1519–1533.

[89] Rampally D. "Iris recognition based on feature extraction". In: (2010).

[90] Kahlil AT and Abou-Chadi FEM. "Generation of iris codes using 1d log-gabor filter". In: *The 2010 International Conference on Computer Engineering & Systems*. IEEE. 2010, pp. 329–336.

[91] Field D. "Relations between the statistics of natural images and the response properties of cortical cells". In: *Josa a* 4.12 (1987), pp. 2379–2394.

[92] Daugman J. "High confidence visual recognition of persons by a test of statistical independence". In: *IEEE transactions on pattern analysis and machine intelligence* 15.11 (1993), pp. 1148–1161.

[93] Martinez M. and Heiner O. *Conditional generative adversarial networks for solving heat transfer problems*. Tech. rep. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2020.

[94] De Marsico M., Nappi M.ichele, and Riccio D. "Is_is: Iris segmentation for identification systems". In: *2010 20th International Conference on Pattern Recognition*. IEEE. 2010, pp. 2857–2860.

[95] Frucci M., Nappi M., Riccio D., and di Baja G. "WIRE: Watershed based iris recognition". In: *Pattern Recognition* 52 (2016), pp. 148–159.