Iris Multi-Frame Super-Resolution with Deep Learning

Ermanno Righini

May 2, 2021

1 Introduction

In the context of iris recognition, an high quality image is essential in order to achieve good performance in the identification process. That is why super-resolution techniques have been studied and applied to improve the accuracy of iris recognition methods. Super-resolution is the class of techniques aimed at increasing the resolution of low quality images. While in the general use case we might just be interested in enhancing the visual quality of the image, in the case of biometrics we want to increase the recognition performance. In the field of biometrics, the literature distinguishes super resolution techniques in reconstruction-based and learning-based. The main difference between the two categories is that reconstruction-based methods extract information from a set of low-resolution images, while learning-based techniques "learn" to generate high resolution images starting from a single image. Both approaches have their own merits and this is why we decided to employ both in this experimental project, that is to learn a mapping from a set of low resolution images to the corresponding high resolution image. This in theory should allow us to both extract all the extra information present in the images and learn the peculiarities of the restricted domain space of iris images to extract more information.

In this project I implemented the 3DWDSR-Net architecture, previously introduced for the problem of multi-frame super-resolution for satellite imaging.

1.1 Error and Accuracy Metrics

In the field of biometry and in particular iris recognition many metrics are used to describe the performance of a system. The most used in literature are False Acceptance Rate (FAR) and False Rejection Rate (FRR). From a security point of view it is important to have a low FAR, because this means that it is very unlikely that someone is misidentified as another person, on the other hand from a user convenience point of view it is desirable to have a low FRR. Usually in an identification system it is possible to choose the desired error tolerance of the system, where an higher tolerance will turn into an higher FAR and lower FRR, and a lower tolerance implies the opposite. The point where FAR and FRR are the same is called Equal Error Rate (EER).

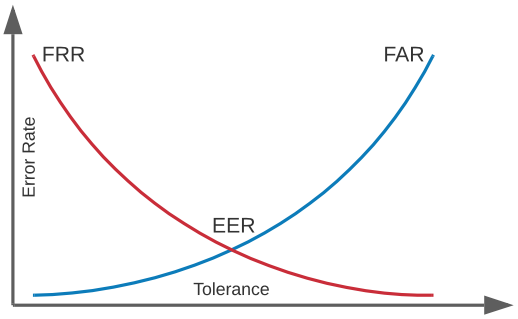

Figure 1: *Recognition Error in relation to Tolerance*

In the field of super-resolution it is instead usual to talk about Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM).

$$MSE = \frac{\sum_{i=1}^{n} y_i - \hat{y}_i}{n} \tag{1}$$

$$PSNR = 10 \cdot log_{10}(\frac{MAX_Y^2}{MSE}) \quad (2)$$

PSNR measures the ratio between the genuine information contained in the image and the extra noise. SSIM describes instead the perceived similarity of two images. It differs from MSE and PSNR, because it considers the relation of locally adjacent pixels, hence the structural nature of the metric.

Lastly in the field of super-resolution are also used perceptual and adversarial accuracy metrics, which are less rigorously defined and try instead to give a more "human-like" interpretation of the similarity between two images.

2 State of The Art

As seen in the survey [1] a lot of research has been done in the field of super resolution for iris recognition. And the research is traditionally divided between reconstruction-based methods and learning-based methods.

As for the reconstruction-based techniques, one of the most recent ones is [2] which first performs iris segmentation to obtain a polar image, then chooses the best frame using the discrete cosine transform, this frame will later be used as a reference point for the alignment of the other frames, which is done using a diamond search algorithm. The reconstruction step is then performed on local patches of the image, in order to account for deformations using Gaussian process regression. This technique produces promising results, with a recognition accuracy of 96.14%, improving from the previous state of the art algorithm's 88.72%.

As for the learning-based methods, [1] presents a technique based on Principal Component Analysis, which improves recognition performance for up to x16 up-sampling and tested recognition performance on 6 different iris comparators with a minimum error of 7.79% in the best case. In the recent years a lot of studies have adapted deep-learning methods to the task of iris super-resolution. For example [3] tried to apply SRCNN [4] and VDCNN [5] models in particular by using transfer learning and thus training the model first on datasets of textures, natural images and testing on a dataset of iris images, showing that transfer learning can be performed successfully even from datasets not focused on iris images. Also showing not much difference in performance for SRCNN and VDCNN models. In [6] the authors tried to understand if photo-realism of the generated SR image is beneficial to recognition performance. To do this they compared three different models: SRCNN, VDCNN and SRGAN [7] and they've seen that in most cases photo-realism isn't related to recognition performance, in particular for the case of SRGAN, the generated images are rich of details, that are hallucinated by the model, worsening the recognition performance. In [8] the same authors tried to add a post-processing step to the super resolution step called Image Re-Projection, where the artifacts in the image are reduced iteratively until a certain threshold is met. In this work the models tested were VDCNN, SRGAN and DCSCN [9]. Image Re-Projection showed some marginal improvements and the model that obtained the best recognition scores was VDCNN. Another recent study [10] aimed at testing the effectiveness of SRGAN in the task of super-resolution for iris recognition and they found out that it often behaves worse than a simple MSE loss model or even bi-cubic interpolation. Finally [11] explored the possibility of mixing the idea of an Adversarial Generative Network with a custom identity preserving loss function, and they found out that the network performed better when both loss functions were used together, achieving a top 5.7% EER in recognition with x8 super-resolved images.

2.1 Video Super Resolution

A very similar problem to Multi-Frame Super-Resolution is Video Super-Resolution, where consecutive frames of a video clip can be seen as the low resolution input frames for the generation of the high-resolution image in a Multi-Frame Super-Resolution problem.

The authors of [12] did a good job of explor-

ing and comparing different techniques for the problem of video super-resolution. They categorized different techniques in "Methods with alignment" and "Methods without alignment". The formers are methods that involve a preprocessing step to align frames, using a specialized module for motion estimation and compensation, that can be learned by the model or given as a prior, while the latter are methods that don't involve any motion compensation, but try to learn the entire process in an end-to-end fashion.

For the problem of super resolution with scale factor 2 and 3 they concluded, using the Vid4 dataset for testing with PSNR and SSIM accuracy metrics, that the best methods were:

- **MMCNN** [13] "Multi-Memory Convolutional Neural Network": The model is subdivided into 5 major modules: an optical flow module for motion estimation and compensation, feature extraction, multi-memory detail fusion, feature reconstruction and upsampling module using a sub-pixel convolutional layer. In particular it uses ConvLSTM modules to merge spatio-temporal information.
- **RRCN** [14] "Residual Recurrent Convolutional Network": The model is a bidirectional recurrent neural network, that uses GLG-TV [15] for motion estimation and compensation, the output of the network is given by the sum of the forward pass, the backward pass and the output of the single-image super-resolution technique EDSR [16].
- **3DSRNet** [17] "3D Super-Resolution Network": This technique doesn't involve any motion compensation, and only relies on 3D convolution to extract high-frequency residual features that are summed to the input upscaled target frame to generate an high-resolution image.

2.2 Multi Frame Super Resolution for Satellite Images

While not strictly connected to the field of Iris super-resolution, I decided to investigate the techniques used in the field of satellite imaging, because there has been a lot of work and research in the recent years, thanks to the PROBA-V competition organized by the European Space Agency. The best performing published techniques, that have been used in the competition are:

- **RAMS** [18]: this is a residual attention model, that makes use of 3D convolutions combined with visual attention in order to obtain an aware data fusion and information extraction from the multiple input frames.
- **3DWDSRNet** [19]: This model is a proposed improvement upon the 3DSRNet architecture, where the 3D Convolutions are replaced with 3D WDSR blocks [20] and the upscaling path is replaced with a 2D WDSR block.
- **HighRes-net** [21]: in this work the authors developed a model that performs recursive fusion of adjacent frames, making the model scalable to a variable amount of input frames. The fused low resolution frame encoding is then super-resolved to an high-resolution image.
- **DeepSUM** [22]: This model is divided in three conceptual parts: shared 2D convolutions for feature extraction, image registration module, and 3D convolution for feature fusion and high-resolution image generation.

3 Technique

The technique I used is mainly based on the 3DWDSRNet architecture [19] which was proposed for the problem of super resolution applied to satellite images. The technique is an enhancement upon the 3DSRNet architecture [17] where the 3D convolutions have been replaced with 3D Wide-Activation Residual blocks as in [20]. In addition to that I also tried to use a perceptual loss function based on the VGG19 model architecture for image classification as was introduced in [23].

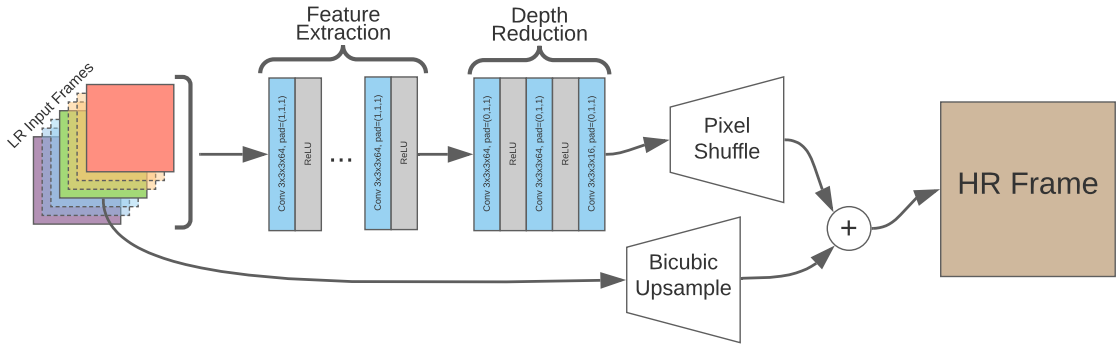

Figure 2: *3DSRNet architecture, as described in the original paper, but with 7 input frames instead of 5.*

In the following subsections I describe all the components of the implemented model architecture.

3.1 3DSRNet

The 3DSRNet architecture arises from the idea that as 2D convolution layers are useful to extract information from the spatial structure of images, 3D convolutions can extend this idea to the time dimension.

As can be seen in Figure 2, the model takes as input a sequence of frames, where the middle frame is the target frame of the super-resolution process. Then the sequence of frames is stacked in a 3D tensor along the depth dimension (for example if I have 7 input frames of resolution 32×32, I obtain a tensor of shape $(7, 32, 32)$).

The input sequence is then passed to a series of 3D convolution layers to extract features from the spatial and temporal dimensions, where it is padded in order to preserve the original image size and temporal depth. The padding along the temporal dimension is especially important because it would be impossible to build a model deeper than a few layers without it.

Then in the last few layers the padding is removed along the temporal dimension to aggregate all the information in a single frame. In order to aggregate all the frames, given an input sequence of length t, it's necessary to perform a number of convolutions equal to:

$$n = \left\lfloor \frac{t}{k-1} \right\rfloor \tag{3}$$

Where k is the size of the convolution kernel. The last aggregation convolution layer must have an output number of channels equal to:

$$n_{channels} = scale^2 \tag{4}$$

This is a necessary prerequisite for the next step that is pixel shuffling.

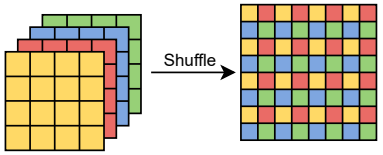

Figure 3: *Pixel shuffling for scale factor 2, notice how the shape changes from* $(4, 4, 4)$ *to* $(1, 8, 8)$*, effectively scaling the image by scale factor* 2

A pixel shuffling operation (also called sub-pixel convolution) is performed as introduced in [24] to reshape a tensor of shape (s^2C, W, D) into a tensor of shape (C, sW, sD), where s is the scaling factor. This is a very efficient alternative to the method of transposed convolution (or de-convolution).

In the case of 3DSRNet the resulting frame from the step of pixel shuffling is just the residual part of the final HR image. To obtain the final high-resolution frame, the input target frame of the sequence, upscaled using the

bicubic interpolation, is added to the residual image. This is done to separate the low-frequency feature extraction (that can be easily achieved with bicubic interpolation) from the high-frequency feature extraction that is instead performed by the learned model, making the problem easier and thus achieving higher accuracy.

3.2 WDSR Block

The WDSR block was introduced in [20] as an improvement to [16], it improves the accuracy of single image super-resolution tasks. As demonstrated in the original paper, this block configuration increases performance while also reducing the number of parameters. This is achieved through the use of 1×1 convolutions which are used for efficient channel expansion and reduction around the non-linear activation functions (ReLUs).

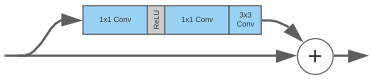

Figure 4: *WDSR Residual Block*

For the case of the 3DWDSRNet architecture, this concept has been extended to 3D convolutions. In the feature extraction part of the 3DSRNet architecture the 3D convolutions have been replaced with 3D WDSR blocks. In the depth reduction part of the architecture the 3D convolutions are instead kept unchanged, since there is a mismatch between the input and output shape of the layers, making the use of 3D WDSR blocks non-trivial.

3.3 Weight Normalization

It is common practice the use of batch normalization [25] in neural networks as a way of regularizing gradients and reducing interval covariance shift, leading to the use of higher learning rates and faster training. But in the case of image super-resolution the use of batch normalization can bring no benefit at all, or even be detrimental to performance as shown in [20]. That's why in this project I used the technique of weight normalization [26], which consists in decoupling the length of the weight vectors from their direction. Given a neuron with output y in the form of:

$$y = \mathrm{w} \cdot \mathrm{x} + b \tag{5}$$

Where x is the layer input vector and w are the respective weights. With weight normalization the w component gets re-parameterized as:

$$\mathrm{w} = g\frac{\mathrm{v}}{|\mathrm{v}|} \tag{6}$$

Where g and v are the new parameters replacing w, respectively representing its length and its direction.

Thanks to weight normalization the converging speed increases and a higher learning rate can be used for learning. In addition to that, for the case of image super-resolution it has been observed that weight normalization leads to an increase of both training accuracy and testing accuracy.

Contrary to batch normalization there is no need of removing the weight normalization re-parameterization during evaluation and inference, since the original formulation and the re-parameterized one are functionally equivalent. Though it is possible to remove it if someone wishes to, by simply recalculating the original weight vector.

3.4 Perceptual Loss

The idea of perceptual loss was first applied to the field of super-resolution in [23], where the authors used it to obtain state of the art results in both style transfer and single image super-resolution.

The main idea is that the loss functions based on single pixel values (like MSE or L1 loss) doesn't provide an error estimate correlated with what a human might perceive. To solve this issue, the error is instead calculated on higher level features extracted both from the predicted image and from the ground truth

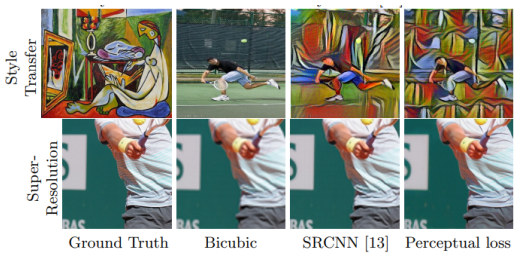

Figure 5: *Example of the use of perceptual loss for the task of style transfer and super resolution (example taken from [23])*

image using a pre-trained neural network, like for example VGG19. [27]

In the original paper, perceptual loss was defined as the squared euclidean distance between the feature representations of two different images. So given the network up to the activation function $\phi(x)$, where j is the layer index of the network, the respective loss function is defined as:

$$l^{j}_{feat}(\hat{y}, y) = \frac{|\phi_j(\hat{y}) - \phi_j(y)|^2}{C_j H_j W_j} \tag{7}$$

In the paper they also implemented a single-image super-resolution model using this features loss formulation, in particular using the features produced by the layer `relu_2_2` of the VGG16 model.

4 Implementation

The models have been implemented using the Pytorch library [28] and using the Weights & Biases platform [29] to track experiments results.

For the purpose of comparison I implemented a few different models architectures:

- **3DWDSRNet**: Very close to the implementation from the original paper [19], but using bicubic interpolation in the upscaling path.
- **3DVGGNet**: Same as **3DWDSRNet** but trained with VGG perceptual loss
- **2DWDSRNet**: Same structure to **3DWDSRNet** but with 2D WDSR blocks and using only the target frame of the sequence.
- **2DSRNet**: Same as **2DWDSRNet** but with simple 2D convolutions instead.
- **Bicubic**: Simple bicubic interpolation for reference
- **Bilinear**: Simple bilinear interpolation for reference

Weight normalization was used in all implementations.

The code for the implementation can be found at:
`https://github.com/righier/iris_mfsr`

4.1 3DWDSRNet implementation

My implementation of 3DWDSRNet follows very closely the original implementation.

In particular I use 8 layers of 3D WDSR blocks with an expansion factor of 6 (as suggested by [20]).

For the depth reduction step I use 3 layers, since my input sequences are composed of 7 frames and the convolution kernel size is 3.

4.2 Upscaling Path

For the implementation of the upscaling path I tried 3 different methods:

- Bicubic interpolation
- Bilinear interpolation
- Custom 2D convolution upscaling module

From these different methods the one that gave me the best results and was the fastest to train was the Bicubic Interpolation. For that reason I decided to use it in all the models I implemented.

4.3 Perceptual Loss implementation

In my implementation I use a variation of the previously explained feature loss from [23]. Instead of using the output of a single layer from the VGG19 model, I use the sum of the feature loss computed for the first 3 activation functions: `relu_1_2`, `relu_2_2`, `relu_3_3`. This is

done as an attempt to use both local and global features for calculating the loss score.

5 Training

All the training was performed on the Google Colab online platform, on virtual machines equipped with Teslas P100 or V100 GPUs.

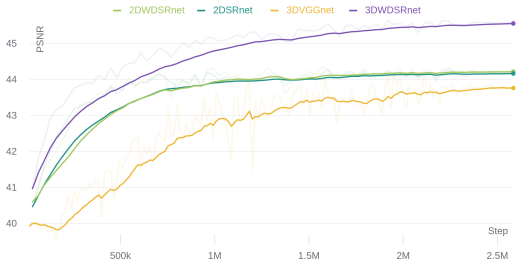

Figure 6: *PSNR over time for all trained models*

I wrote a training loop using the Pytorch API, where I simultaneously perform training on the training dataset and perform evaluation on the validation dataset. This allows me to see if the model is over-fitting by simply checking if the training loss keeps decreasing while instead the validation loss increases.

I also defined some checkpoints along the training procedure which allow me to save the model with the best accuracy on the testing dataset.

The heaviest model (**3DVGGNet**) took approximately 12 hours to train on a Tesla P100 while the simpler **2DWDSRNet** took only 2 hours.

For all the models, the Adam optimizer was used for updating the parameters, and the weights were initialized with the Xavier initialization formula.

6 Learning Rate Scheduling

To vary the learning rate over time and achieve better accuracy faster, I decided to use the schedule proposed by [30]

With maximum learning rate set to 5×10^{-4} and minimum set to 10^{-5}.

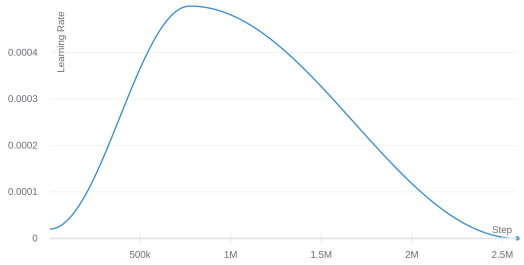

Figure 7: *One Cycle Learning Rate Schedule*

This schedule proved empirically to be way more effective than the classical exponential decaying learning rate. It allowed to train the model much faster, training for only 5 epochs per model and even obtaining better accuracy.

7 Dataset

Due to the lack of datasets with a coupling of high and low resolution images or videos, the most obvious thing to do is to build a dataset synthetically by performing simulated down-sampling and generating the low resolution images starting from the high resolution ones. In order to apply the model in the real world it is important that these degrading transformations are general enough to cover the case of real cameras. These transformations contain: down-sampling, blurring, warping, noise.

Ideally the training dataset would be composed of only iris video sequences, but I could only find one dataset containing iris videos, that is the LAVI DB2 iris database [31], which contains a total of 212 video sequences extracted from 53 different people. Sadly this is not enough diversity to properly train a deep neural network, so I decided to use this dataset for testing purposes while perform training on the Vimeo90k dataset [32] which consists in 91701 7-frame sequences extracted from 39000 video clips.

7.1 Data preprocessing

In order to use the Vimeo90k dataset for the purpose of iris super-resolution, all video se-

quences were transformed to grayscale images, since iris images are usually captured with near-infrared cameras which produce grayscale images. Then the central image of the 7-frame sequence was chosen as the target high-resolution image and all the frames were blurred and downscaled to obtain their low resolution counterparts.

To enhance the model performance I decided to apply some preprocessing steps, such as data standardization and patch extraction. Standardization consists in calculating the training dataset's mean and standard deviation and then applying the transform:

$$x' = \frac{(x - \hat{x})}{\sigma} \tag{8}$$

This step turns the dataset's mean to 0 and the standard deviation to 1, which makes gradient descent converge faster.

Patch extraction consists instead in subdividing the input image in smaller patches, in order to increase the batch size and variability during training. By applying patch extraction, I extracted 8 patches from each video sequence creating a new dataset of individual patches, consisting of 733608 training examples.

I also don't perform any sort of data augmentation since the training dataset is already big enough. This suggests that it could be possible to train a model of equivalent accuracy by using a smaller dataset in conjunction with data augmentation techniques, such as random cropping, flipping and rotation.

Another direction to explore is the use of "crappification" techniques, such as adding noise or blur to the input frames. This should lead to models that are more robust, and more applicable to the real world, but it is also harder to achieve the same kind of accuracy since the model is forced to learn both the task of denoising, de-blurring and super-resolution.

Lastly, I decided to not use any kind of image registration or re-projection in opposition to the original implementation of the method [19] where the authors decided to perform image registration using phase cross-correlation, which works well in the original domain of satellite images since the input frames differ only in light exposure and position shift, but in the case of iris images non-rigid deformations are more prevalent, making the use of phase cross-correlation not beneficial.

For the LAVI DB2 dataset I instead decided to extract 5 7-frame sequences at random from each of the 53 subjects, producing a total of 265 video sequences.

8 Experimental Results

For testing I decided to focus on super-resolution with scaling factor 4, since with higher scale factors it is very hard to extract any meaningful information and with lower factors the problem becomes trivial, so all the experimental data in this report is referring to experiments conducted with scaling factor 4.

Table 1: *Models sorted by PSNR*

Model	PSNR
3DWDSRNet	**45.643**
2DWDSRNet	44.344
2DSRNet	44.288
3DVGGNet	44.163
Bicubic	40.449
Bilinear	39.154

As can be seen from Table 1 the best performing model according to the PSNR metric is **3DWDSRNet** which shows that the model was able to extract information from the extra frames, leading to an higher score compared to the single-frame models (2DWDSRNet, 2DSRNet).

From a visual analysis of the results (Figure 8), it can be seen that all the models performed much better than the baseline (bicubic and bilinear interpolation), and visually the results of the models are very similar, which is impressive since **2DWDSRNet** and **2DSRNet** make use of a single frame.

Between the two single-image models, there isn't much difference visually, but **2DWDSRNet** still managed to achieve better accuracy

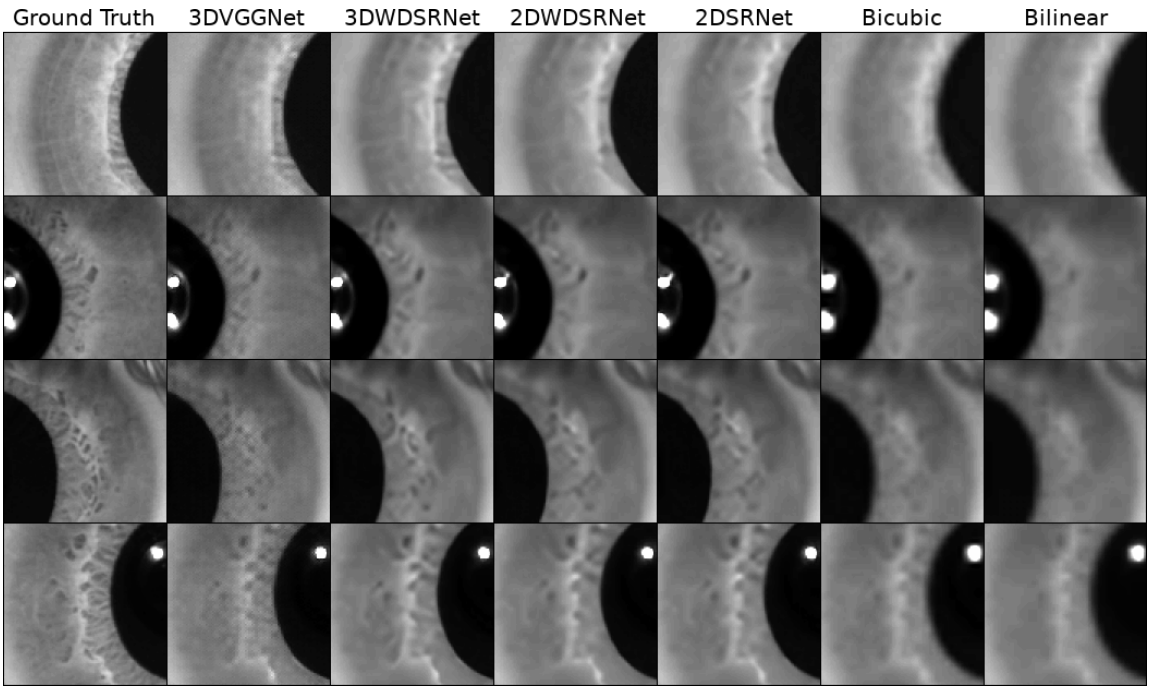

Figure 8: *Iris patches super resolved with scaling factor 4 by all the implemented methods (the colors have been corrected for better visualization)*

than **2DSRNet**

Lastly, the results produced by **3DVGGNet** look the most detailed, but at the same time suffer from checkerboard artifacts, which are a result of the combination of pixel shuffling and perceptual loss.

8.1 Iris Recognition

To test the efficacy of the trained model I tried to check with an iris recognition framework if the trained models led to an higher recognition accuracy compared to the baseline methods, like bicubic and bilinear interpolation.

To do so I used the framework USIT [33] to perform recognition by using the CAHT [34] iris segmentation algorithm, the 1D Log Gabor filter [35] for feature extraction and Hamming distance between codes for matching.

In more detail, I extracted 5 frame sequences randomly for each subject in the LAVI DB2 database [31], then I calculated the EER values in Table 2 for each upscaling method by comparing all samples with each other. I performed two experiments for each method, one were the upscaled images were of width 512 pixels and one where they were of width 256.

Table 2: *EER values for different methods for upscaling iris image sequences*

Model	EER (512px)	EER (256px)
Ground Truth	**4.71%**	5.51%
Bicubic	6.73%	3.92%
Bilinear	8.46%	10.06%
2DWDSRNet	7.30%	4.30%
3DWDSRNet	6.49%	7.12%
3DVGGNet	10.30%	**2.60%**

I think these EER values are not correlated with the detail extracted by the super-resolution methods, since for the tests with images of width 256 pixels it would not make sense for the ground truth images to perform worse than upscaled images. What I think happened in this experiments is that the original images are a too noisy, and the recognition soft-

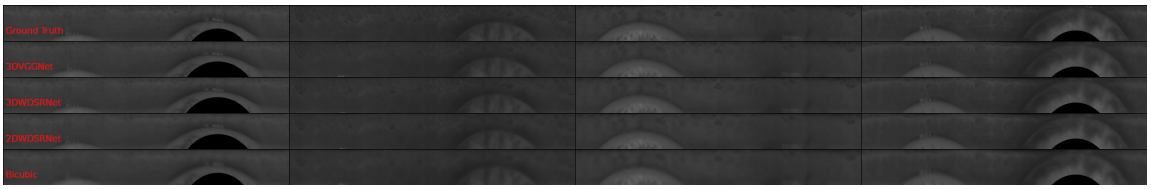

Figure 9: *Iris textures extracted with CAHT algorithm from upscaled images using different methods (ground truth is the texture extracted from the high resolution image)*

ware can perform better when the images are smoothed from the upscaling methods.

If we instead look at the segmentation output textures it is visually apparent how the proposed super-resolution methods produce higher quality textures compared to the baseline methods, and how the multi-frame variants of the model perform better compared to the single-frame ones.

9 Conclusions

These experimental results show that even with a relatively simple model architecture, that doesn't perform any motion compensation, it is possible to achieve very promising results in the field of multi-frame iris super-resolution, by training for a reasonably short amount on time on freely available hardware resources (Google Colab).

Another observation is that classical metrics for image similarity cannot properly evaluate the quality of generated images, as seen for the case of the **3DVGGNet** model, that is scored lower than **2DWDSRNet** while it visibly contains richer high-frequency details. A much more appropriate metric for accuracy would be the error recognition rate of an iris recognition system.

9.1 Future Work

An useful expansion on this project would be to test these results with an iris recognition system to verify that the upscaled images actually contain useful information for the recognition process and therefore improve the recognition performance.

It would be interesting to explore further the idea of Perceptual Loss, since in other works it's use led to significantly better performance.

The first very easy improvement might be to train the Perceptual Network from scratch on a dataset composed of only grayscale images, since all the current state of the art networks are trained instead on RGB images. This could lead to increased performance since the network would be forced to extract information from the structural properties of the image instead than color information. Different model architectures could also be tried for the Perceptual Loss Network, since the VGG architecture was first presented in 2014 and since then a lot of more performant end efficient architectures have been proposed.

On a different direction it might be interesting to twist the idea of Perceptual Loss, and instead use a neural network pre-trained to perform iris recognition. This in theory should optimize the super-resolution model for the extraction of features important directly to the iris recognition problem. The problem with this approach is the need of an iris video dataset big and diverse enough to train an iris recognition model from scratch.

It would also be nice to try training the model on a perturbed dataset, with noise, motion blur and gaussian blur, to make it more resiliant to these kinds of artifacts. As shown by [36] where they performed a "crappification" preprocessing, consisting of altering input images with salt-and-pepper noise and Gaussian additive noise. This led to increased generality and transfer learning capabilities of the model on never seen before datasets.

A different aspect that could be improved

is the robustness towards pixel-shifts between input frames. The current architecture is able to work with small pixel-shifts but a dedicated motion estimation and motion compensation module would be helpful for more challenging cases.

Another possible improvement to the system would be to add a "best-frame" selector module, as a filtering step before feeding a frame sequence to the model. This module could check image clarity and iris visibility to perform the heavy and slow super-resolution process only on the best frames of the video.

References

[1] Fernando Alonso-Fernandez, Reuben A Farrugia, Josef Bigun, Julian Fierrez, and Ester Gonzalez-Sosa. A survey of super-resolution in iris biometrics with evaluation of dictionary-learning. *IEEE Access*, 7:6519–6544, 2018.

[2] Anand Deshpande and Prashant P Patavardhan. Super resolution and recognition of long range captured multi-frame iris images. *IET Biometrics*, 6(5):360–368, 2017.

[3] Eduardo Ribeiro and Andreas Uhl. Exploring texture transfer learning via convolutional neural networks for iris super resolution. In *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5. IEEE, 2017.

[4] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[5] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[6] Eduardo Ribeiro, Andreas Uhl, and Fernando Alonso-Fernandez. Iris super-resolution using cnns: is photo-realism important to iris recognition? *IET Biometrics*, 8(1):69–78, 2019.

[7] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[8] Eduardo Ribeiro, Andreas Uhl, and Fernando Alonso-Fernandez. Super-resolution and image re-projection for iris recognition. In *2019 IEEE 5th International Conference on Identity, Security, and Behavior Analysis (ISBA)*, pages 1–7. IEEE, 2019.

[9] Jin Yamanaka, Shigesumi Kuwashima, and Takio Kurita. Fast and accurate image super resolution by deep cnn with skip connection and network in network. In *International Conference on Neural Information Processing*, pages 217–225. Springer, 2017.

[10] Koji Kashihara. Iris recognition for biometrics based on cnn with super-resolution gan. In *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, pages 1–6. IEEE, 2020.

[11] Yanqing Guo, Qianyu Wang, Huaibo Huang, Xin Zheng, and Zhaofeng He. Adversarial iris super resolution. In *2019 International Conference on Biometrics (ICB)*, pages 1–8. IEEE, 2019.

[12] Hongying Liu, Zhubo Ruan, Peng Zhao, Chao Dong, Fanhua Shang, Yuanyuan Liu, and Linlin Yang. Video super resolution based on deep learning: A comprehensive survey. *arXiv preprint arXiv:2007.12928*, 2020.

[13] Zhongyuan Wang, Peng Yi, Kui Jiang, Junjun Jiang, Zhen Han, Tao Lu, and Jiayi Ma. Multi-memory convolutional neural network for video super-resolution. *IEEE Transactions on Image Processing*, 28(5):2530–2544, 2018.

[14] Dingyi Li, Yu Liu, and Zengfu Wang. Video super-resolution using non-simultaneous fully recurrent convolutional network. *IEEE Transactions on Image Processing*, 28(3):1342–1355, 2018.

[15] Marius Drulea and Sergiu Nedevschi. Total variation regularization of local-global optical flow. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 318–323. IEEE, 2011.

[16] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. *CoRR*, abs/1707.02921, 2017.

[17] Soo Ye Kim, Jeongyeon Lim, Taeyoung Na, and Munchurl Kim. 3dsrnet: Video super-resolution using 3d convolutional neural networks. *arXiv preprint arXiv:1812.09079*, 2018.

[18] Francesco Salvetti, Vittorio Mazzia, Aleem Khaliq, and Marcello Chiaberge. Multi-image super resolution of remotely sensed images using residual attention deep neural networks. *Remote Sensing*, 12(14):2207, 2020.

[19] Francisco Dorr. Satellite image multi-frame super resolution using 3d wide-activation neural networks. *Remote Sensing*, 12(22):3812, 2020.

[20] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018.

[21] Michel Deudon, Alfredo Kalaitzis, Israel Goytom, Md Rifat Arefin, Zhichao Lin, Kris Sankaran, Vincent Michalski, Samira E Kahou, Julien Cornebise, and Yoshua Bengio. Highres-net: Recursive fusion for multi-frame super-resolution of satellite imagery. *arXiv preprint arXiv:2002.06460*, 2020.

[22] Andrea Bordone Molini, Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Deepsum: Deep neural network for super-resolution of unregistered multitemporal images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3644–3656, 2019.

[23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[24] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient subpixel convolutional neural network, 2016.

[25] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[26] Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *CoRR*, abs/1602.07868, 2016.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch, 2017.

[29] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[30] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019.

[31] Jones Mendonça de Souza and Adilson Gonzaga. Human iris feature extraction under pupil size variation using local texture descriptors. *Multimedia Tools and Applications*, 78(15):20557–20584, 2019.

[32] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8):1106–1125, 2019.

[33] Christian Rathgeb, Andreas Uhl, Peter Wild, and Heinz Hofbauer. Design decisions for an iris recognition sdk. In Kevin Bowyer and Mark J. Burge, editors, *Handbook of Iris Recognition*, Advances in Computer Vision and Pattern Recognition. Springer, second edition edition, 2016.

[34] Christian Rathgeb, Andreas Uhl, and Peter Wild. *Iris biometrics: from segmentation to template security*, volume 59. Springer Science & Business Media, 2012.

[35] A. T. Kahlil and F. E. M. Abou-Chadi. Generation of iris codes using 1d log-gabor filter. In *The 2010 International Conference on Computer Engineering Systems*, pages 329–336, 2010.

[36] Linjing Fang, Fred Monroe, Sammy Weiser Novak, Lyndsey Kirk, Cara R Schiavon, B Yu Seungyoon, Tong Zhang, Melissa Wu, Kyle Kastner, Alaa Abdel Latif, et al. Deep learning-based point-scanning super-resolution imaging. *Nature Methods*, 18(4):406–416, 2021.