3D Super Resolution using Auto Encoders for Face Recognition

F.F. Bulten

***Abstract*—This paper proposed a method to super resolve 3 dimensional facial images using auto encoders such that face recognition can be performed. First different architectures will be tested and the best performing architecture will tested against a simple implementation of interpolation, by performing error analysis and facial recognition. Based on the results the performance of the auto encoder is impressive and better when comparing it to interpolation, facial recognition can be performed reliably and visually the images look better.**

I. Introduction

In recent years methods have been proposed for 3 dimensional face recognition in [9] to name one, this method uses high quality images taken with expensive cameras, under which the Minolta vivid 910. However, more widely available and cheaper 3d cameras, like the Microsoft Kinect and the Intel Realsense, have lower resolution and more noise. That is why methods have been proposed to improve image quality using super resolution in [1], [4] and [3], implementing reconstruction type super resolution. This method comes with the drawback that it is unable to reproduce higher frequency components of the image. Learning based methods are able to reproduce these details using generative adversarial networks or auto encoders resulting in a higher quality image. So, this paper proposes a method using auto encoders to super resolve low resolution depth images images such that face recognition can be performed.

A multitude of methods for 2 dimensional super resolution using auto encoders have been proposed in [10], [7] and [12]. This paper aims to extent this into 3 dimensions by taking low quality depth images of faces and super resolving them into higher resolution, with the expectation that higher frequency components of the image also can be reproduced and facial recognition performance can be improved.

High quality images taken from the Face Recognition Grand Challenge (FRGC) v2 database will be down scaled in 3 dimensions and some noise and outliers will be added, from which the auto encoders will tested and trained. Multiple architectures will tested from where the best one will be chosen and compared to a simple implementation of interpolation in therms of performance, computational complexity and if face recognition can be performed reliably. In addition, the auto encoder will be tested using the FR3DK dataset [6], which is are images taken with a Microsoft kinect camera.

In section II a general overview of super resolution will be given and more elaborate overview will be given of recent the state of the art. Section III will elaborate farther on the preprocessing, auto encoder architectures that will be tested, the interpolation method, how the face recognition will be done and test that will be done using the FR3DK dataset. Section IV will go over the results and finally in section V and VI will present the discussion and conclusion.

II. Related Work

Super Resolution has been around for a for a few years, and three main methods are being used for this, the first one is based in interpolation. HR (High Resolution) images are being reproduced from LR (Low Resolution), by estimating pixels by using interpolation methods such as bi-linear and bi-cubic interpolation, these methods are simple, but it is common that they produce blurry images and are unable to reproduce high frequency components of the HR image. Another method is reconstruction, this methods makes use of prior knowledge knowledge of the image, such as edges [3] or multiple LR images [4]. This method generally performs better than interpolation, however is still limited in the reconstruction as there are steps where interpolation is used and so reproduction of higher frequency components is limited. A third method is based on deep learning, it is assumed that high frequency details from images can be reproduced with either super-resolution generative adversarial networks (SRGAN) [2] or auto encoders [7]. HR and LR images are used to train these networks, these methods seems to produce the best better results then methods mentioned previously. However, are only implemented in 2 dimensions for now.

A. 2D super resolution with auto encoders

Several super resolution methods for 2D images using auto encoders have been presented in recent years, [10] and [7] both propose methods for single image super resolution using coupled deep auto encoders, in which the codes of the low resolution encoder and the high resolution decoder are coupled together via mapping. In addition [7] adds a stacked sparse denoising auto encoder to make the setup noise resilient as well. More recently in [12], it has been proposed to use variational auto encoders for single image super resolution. It does this by combining two variational auto encoders with gaussian mapping, as of now this method produces state of the art results and the best results the writer knows of.

B. 3D super resolution

One simple method for 3d super-resolution, is by doing bi-linear or bi-cubic interpolation and then filtering to get the depth resolution higher, this method is really simple but results in blurry edges. Better methods have been proposed to super resolve depth images using reconstruction [1], [4] and [3], these methods have much better results than simple interpolation. [1] Uses low resolution 3d images from a Microsoft Kinect camera of which the faces first get cropped out of the full image then aligned to a reference image, lastly an higher resolution is approximated using 2d box splines. In [4] images are aligned at first as well, but this method has the addition of noise removal by labeling processing, After which reconstruction type super resolution is applied. Lastly in [3] a Canny edge detector is applied to the low resolution 3d depth image and the high resolution colour image. In the depth image holes are detected and filled, then the depth image is up sampled, local edge enhancement is applied and a filter is applied to enhance the high resolution depth image. These methods produce state of the art results but still have their limitations for reasons mentioned before.

III. Method

This paper will present a method to super resolve 3d images using auto encoders, with the aim to reproduce high quality face images from low quality ones. This is done using depth images as not much has to change in the auto encoder when using depth images compared to normal images because the depth images are essentially 2d images with floating point values instead of integers. For the depth images the FRGCv2 database [5] is used, this database consists of 4950 images of 536 different subjects taken with a minavolta vivid 910, this database is used for training (4155 images of 425 subject), validation (652 images of 101 subjects) and testing (143 images of 10 subjects). For testing the FR3DK data base [6] is used as well, which contains scans of 16 different subjects using different expressions, however, the expressions in the FRGCv2 database are mostly neutral so mostly neutral expressions from the FR3DK dataset will be chosen for testing. The scans are taken with a microsoft kinect sensor. The images are formatted as point clouds so first have to be converted and aligned into proper depth images.

For implementing the auto encoder the keras API for python is used as it is easy to use and has plenty of tools available to make auto encoders, in addition the Plaidml framework can be used to make use of opencl GPUs for training as opposed to CUDA as the latter is not available to the writer.

A. Preprocessing Data

The preprocessing of the data aims to lower the image quality and mimic the performance of an low cost 3d sensor such as the Microsoft kinect, of which certain specifications can be found in [8]. At a distance of 1 meter the depth resolution is about 3mm, so this is also chosen to quantize the HR images at to decrease the depth resolution. In addition the XY-resolution is also decreased by half from 195×165 to 97×82, Gaussian noise with mean 0 and standard deviation 4mm and outliers with probability of 1 in 39 are added. This results in a low quality depth image of a face that can be used to train the auto encoder.

For testing different standard deviations for the noise will be used, these images can be seen in appendix section B in figure 14.

B. Architecture

Different auto encoder architectures will be tried to see what would give the best performance while still being manageable in therms of training times. The main split has been made between fully convolutional auto encoders and normal auto encoders, a fully convolutional auto encoder misses the dense layers from the normal auto encoder. This can be done since the goal is not to compress the data and the only thing that matters is the output and not compressing the data might result in overall better reconstructions. In addition, using a fully convolutional auto encoder saves training time compared to the regular auto encoder as can be seen later in section IV. These auto encoders will be trained over 40 epochs using a batch size of 8, the Adam optimizer and mean squared error as the loss function, the training data will be shuffled each epoch.

1) Regular Auto encoders: Different regular auto encoders will be tried out, with depths of 2, 3 and 4, a code size of 500 will be used, the full architectures can be seen in appendix section A in figures 8, 9 and 10.

2) Fully Convolutional Auto encoders: A similar thing is done for the fully convolutional auto encoders, they have a depth of 2, 3 and 4, the full architectures can be seen in appendix section A in figure 11, 12 and 13.

These auto encoders will be tested using the testing data (FRGCv2) after which the reconstructed image will be compared to the original image, an error map will be made of which the mean, maximum and standard deviation will be determined. Doing this for all the testing images and than taking the mean for each of those will results in the overall performance for each architecture from which the best architecture will be determined.

Once this is done the chosen depth will be adjusted further, in therms of strides and kernel sizes and training parameters such as initial learning rate and batch size will also be adjusted accordingly, to achieve the best performance. The same error analysis is done for this as previously to determine the better architecture.

Once a architecture is chosen will be trained for 200 epochs using the chosen batch size and initial learning rate and optimizer. Then the same error analysis will be done again as before, however now only for the middle part in order to compare it to the interpolation method as well as the use different noise levels, with a standard deviation of 0mm, 2mm, 4mm and 6mm.

C. Interpolation

As control test a simple implementation of interpolation will be used, first an median filter is applied to the low resolution image with a kernel of the size 3×3, after that a Gaussian filter is applied with a kernel size of 3×3, than cubic interpolation is applied to upscale the image to its original size (195×165) after which two median filters are applied each with kernel size of 5×5. Than the same will be done as before, by making an error map with the original image and determining the mean, maximum and standard deviation, however this time only the

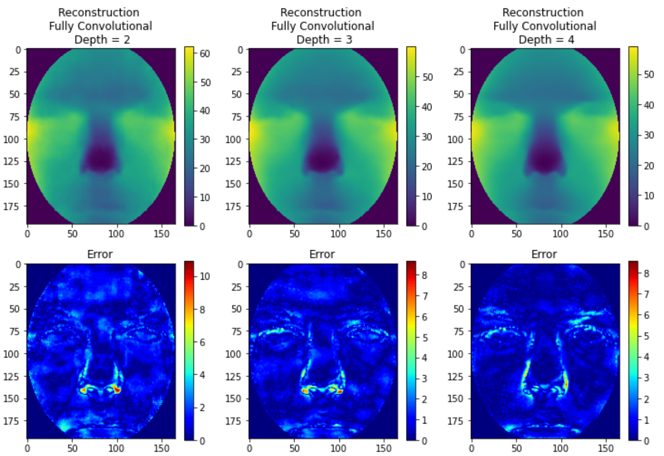


Figure 1: Reconstructions using the fully convolutional architectures (mm)

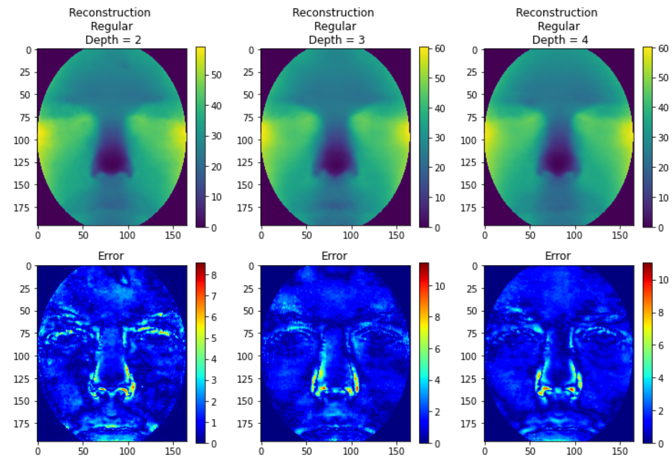


Figure 2: Reconstructions using the regular auto encoder architectures (mm)

middle part of the face is taken, since this method causes big errors at the edges of the face which do not matter as much when face recognition is applied.

D. Face Recognition

Then, using the interpolation method and the auto encoder, 10 images of each of the 10 testing subjects from the FRGCv2 database will be reconstructed such that face recognition [9] can be performed. The standard deviation of the noise will be 4mm, a threshold of 8 will be chosen and the sets of reconstructions and low quality images will be compared to themselves and the original images, from there the false acceptance and false rejection will be determined, which should give a good indication what the performance is like for face recognition.

E. Kinect Images

The Kinect images from the FR3DK database [6], will also be reconstructed using the auto encoder and interpolation, these reconstructions will be compared visually as no real error analysis can be performed. The FR3DK dataset contains point clouds, these first have to be converted and aligned such that the facial features are in the same place as the training data.

IV. Results

In this section the results will be discussed, such as what architecture has been chosen and what performance can be achieved using this architecture. And how does it compare to interpolation and can face recognition be performed reliably by doing the experiments mentioned in section III. For visualisation one image is used for which the original can be seen in appendix section B.

A. Architecture

First off the best performing architecture was determined, the preprocessing was applied on the testing images and the reconstructions were made resulting in the images and error maps seen in figure 1 and figure 2, full sized images can be found in appendix section C.
The mean of the maximums, means and standard deviations when reconstructing the 143 images in the testing set can be seen in table I, as well as the training times when using the plaidml framework with an AMD RX480 GPU. As can be seen in the table and the figures fully convolutional auto encoders performed better as the depth increased, for the regular auto encoders it is the other way around with the best performance coming from the most shallow auto encoder. However the best performance still came from the fully convolutional architectures so it was decided to go on with a depth of 4, as it had the overall best performance and the training times were still manageable.
Next, the auto encoder was adjusted such that better performance can be achieved which resulted in the architecture seen in figure 4, different training parameters were tried as well, but it was found that original parameters were good options already, so a batch size of 8 and the Adam optimizer with the standard initial learning rate of 0.01 an mean squared error as loss function.
The auto encoder consist 4 convolutional 2d layers, each of them has a kernel of with a size of 3x3, a stride of 1 and zero padding to keep the size the same. Two maxpooling layers with the size of 2x2 are added as well to reduce the size with a factor of 4. Than 5 transposed 2d convolutions are used, the first 3 all have a kernel size of 3x3, a stride of 2, and zero padding. Since the image is too big at this point, cropping has to be applied after which there are 2 more transposed 2d convolutional layers with a kernel size of 3x3, a stride of 1 and zero padding. The main differences between this architecture and the architecture used before is that size at the deepest layer is nearly double the size, $25 \times 21 \times 256$ instead of $13 \times 11 \times 256$ and smaller strides are used in the decoder part of the auto encoder, meaning that the kernel sizes can be reduced as well. However, since the resolution of the original picture (195×165) is not perfectly divisible by 8 cropping has to be used. This architecture has been trained for 40 epochs which results in figure 3 and the mean of the

Table I: Mean error, Maximum error, STD of error and training times

σ = 4mm depth step = 3mm	Mean	Maximum	STD	Training Time
Convolutional depth = 2	0.929mm	10.4mm	0.926mm	11 Minutes
Convolutional depth = 3	0.624mm	9.70mm	0.720mm	33 Minutes
Convolutional depth = 4	0.578mm	9.59mm	0.704mm	120 Minutes
Regular depth = 2	0.755mm	10.1mm	0.835mm	13 Minutes
Regular depth = 3	0.862mm	10.6mm	0.961mm	36 Minutes
Regular depth = 4	0.876mm	10.5mm	0.960mm	150 Minutes

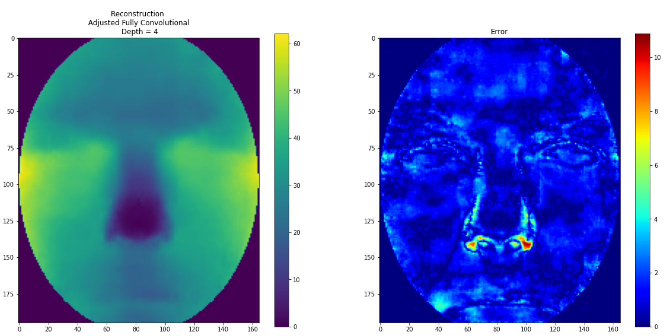


Figure 3: Reconstructions using the adjusted auto encoder architecture (mm)

means, maximums, standard deviations when reconstructing the entire test set can be seen in table II, full size images can be seen in appendix section C. The mean is worse, however the maximums and the standard deviation are a lot better than than the other architecture and the training time is shorter too. The reconstruction itself does look worse however, but when it is trained for longer it does get better as seen later in this paper. So this architecture was trained for 200 epochs which took 7 hours after which not much improvement was made and it will be used from now on.

B. Interpolation vs Auto encoder

The auto encoder will be compared to interpolation method as described in section III, from there all 143 testing images will be up scaled using both methods and for error analysis the images will be cropped such that the edges of face are not taken into account, which contain big errors of about 30mm but these do not matter as much for face recognition performance. The results from one image can be seen in figures 5a, 5b, 6a and 6b with the means of the errors, maximums and standard deviations of the entire testing set

Table II: Mean error, Maximum error and STD of error for the adjusted auto encoder

σ = 4mm depth step = 3mm	Mean	Maximum	STD	Training Time
Adjusted Architecture	0.687mm	5.24mm	0.650mm	96 mins.

Table III: Means, Maximums and STDs of the auto encoder and interpolation for different noise levels

σ = 0mm depth step = 0.03mm	Mean	Maximum	STD
Auto Encoder	0.707mm	7.74mm	0.583mm
Interpolation	0.573mm	8.27mm	0.663
σ = 2mm depth step = 0.03mm	Mean	Maximum	STD
Auto Encoder	0.593mm	7.86mm	0.586
Interpolation	0.699mm	8.45mm	0.710
σ = 4mm depth step = 0.03mm	Mean	Maximum	STD
Auto Encoder	0.597mm	8.20mm	0.627mm
Interpolation	0.960mm	8.64mm	0.831mm
σ = 6mm depth step = 0.03mm	Mean	Maximum	STD
Auto Encoder	0.892mm	9.59mm	0.798mm
Interpolation	1.26mm	9.41mm	1.01mm

seen in table III. Even though the auto encoder is only trained with Gaussian noise with a standard deviation of 4mm, it is still able to reproduce good images at other noise levels. For σ = 2mm, the reconstruction is is even better than the reconstruction for σ = 4mm however, not by much as seen in the table. For σ = 0mm the maximum and STD got even better, but the mean is worse compared to higher sigmas, except for σ = 6mm for which everything got worse. And when comparing the auto encoder to the interpolation method, is can be seen that overall the auto encoder is better, only losing out in mean error for $\sigma = 0$ and maximum for $\sigma = 6mm$, and when looking at the images it can be seen that the auto encoder reproductions are sharper and have a lot more detail than the reconstruction from the interpolation, however for the reconstruction with $\sigma = 6mm$ it can be seen that the nose is skewed this was found for more images. The interpolation reconstruction still contain a lot of noise and for $\sigma = 0mm$ and $\sigma = 2mm$ the quantization steps can still be seen in the image. Execution times for the interpolation method when reconstructing all 143 images in the testing set was much better, 0.054 seconds compared to the 1.7 seconds that the auto encoder took to do the same thing, but this is still not so long that it is unusable.

C. Face recognition

Then the reconstructions were tested using facial recognition software [9], this is done as described in section III and the FRR (false reject rate) and FAR (false acceptance rate) can be seen in table IV, the false acceptance and false reject for each subject can be seen in appendix section E. When two sets are compared a total of $100 \times 100 = 10000$ comparisons are done, except for the interpolation set since one image got corrupted there were only 99 images in the set so for interpolation vs high quality there were $99 \times 100 = 9900$ comparisons and for interpolation vs interpolation there were $99 \times 99 = 9801$ comparisons. It must also be noted that when a set is compared to itself the doubles are counted as well.

As can be seen in the table the reconstructions with the auto encoder perform the best, then the interpolation and worst are the low quality images. When the low quality (LQ) images and the interpolation (IP) images are compared to

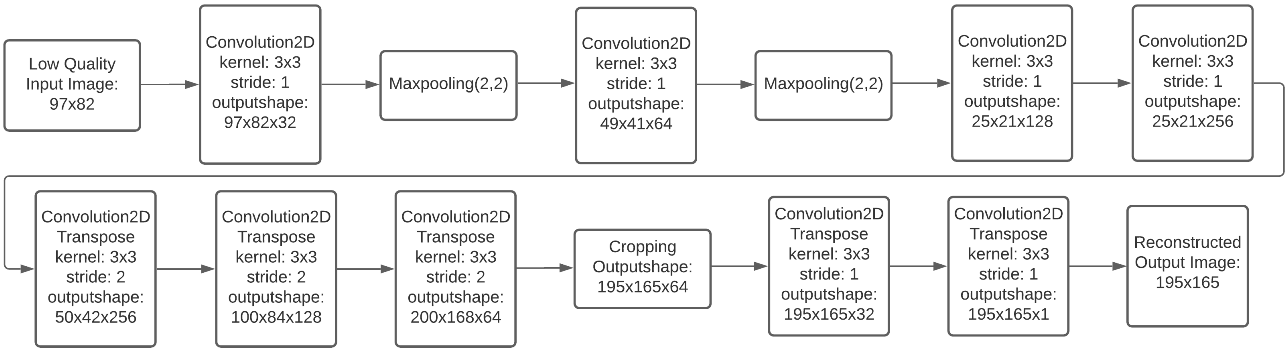
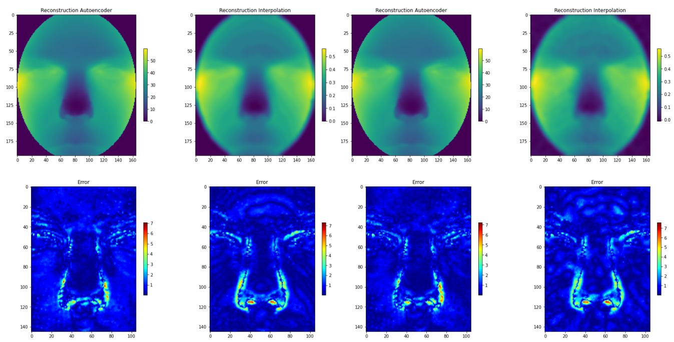
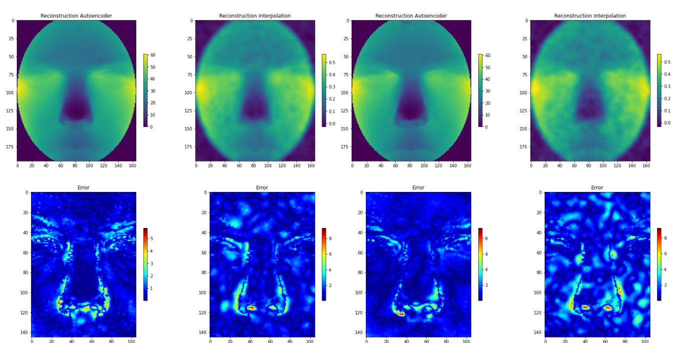


Figure 4: Final auto encoder architecture



(a) σ = 0mm (b) σ = 2mm

Figure 5: Reconstructions and errors for different noise levels



(a) σ = 4mm (b) σ = 6mm

Figure 6: Reconstructions and errors for different noise levels

the high quality (HQ) images it can be seen that the FRR is high, as images are not recognisable, the interpolation did improve it compared to the LQ images but performance is still quite poor. When compared to themselves the IP images have a much lower FRR, however the FAR is much higher so performance is still quite poor, this can be explained by the fact that the interpolation method blurs the images, which makes the different subjects look like each other. Low Quality compared to its self still gives quite poor results, however this is more balanced in therms of FRR and FAR. The auto encoder comparisons are much better than the other two, both the FRR and the FAR are low suggesting that the auto encoder

Table IV: FRR and FAR for all comparisons

Auto encoder vs Original image		Auto encoder vs Autoencoder	
FRR	FAR	FRR	FAR
0.041	0.0013	0.049	0.003
Interpolation vs Original Image		Interpolation vs Interpolation	
FRR	FAR	FRR	FAR
.0.884	0	0.050	0.80
Low qualtiy vs Original image		Low quality vs Low quality	
FRR	FAR	FRR	FAR
0.999	0	0.282	0.111
Original image vs Original image			
FRR		FAR	
0.024		0.00044	

does its job well, but it is still not on par with the original images as they have a much lower FRR and FAR. One thing to note is that for the first subject, one image only gave false negatives for both the auto encoder reconstructions and the original images. However when looking at the image it must be concluded that it is the same subject even though the negative results from the face recognition software suggesting that it is an error within the software causing high false rejects for the first subject.

D. Kinect Images

Lastly the auto encoder and interpolation method are tested with the Microsoft Kinect data, the results can be seen in figure 7. In this case the original image is of the size 195×165 so this first has to be down scaled to 97×82 since that is the only size the auto encoder accepts at its input. Then this image gets reconstructed with the auto encoder and the interpolation method, it can be seen that the reconstruction by auto encoder improves image quality significantly. In the original image the steps due to limitation in depth resolution are still visible but these are completely gone in the reconstruction with the auto encoder, some of the details near the eyes and mouth are reconstructed as well resulting in an overall better image. The interpolation also does improve the depth resolution, but the image is blurry and the reconstructions are not as good as the ones the auto encoder makes as in some places the steps are still visible and the odd shapes of the nose are not gone. The auto encoder is able to make better shaped noses, however these are still a bit skewed in the second and third image.

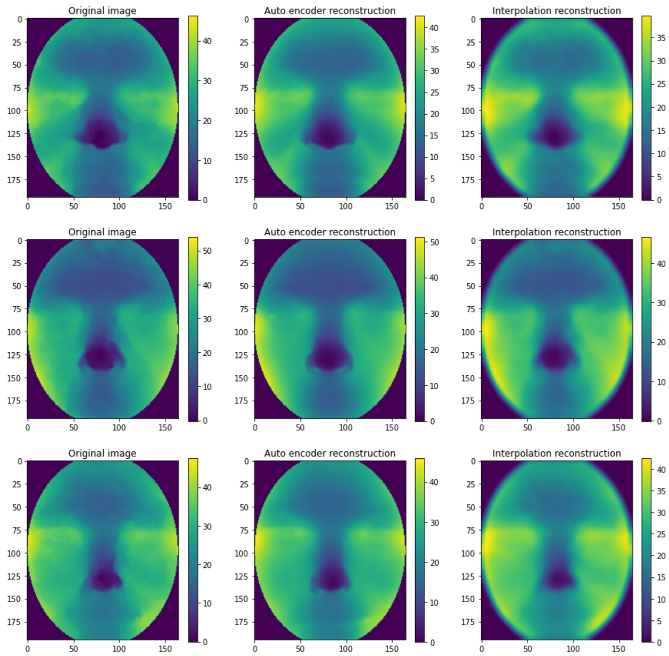


Figure 7: Reconstructions with for the kinect images (mm)

V. Discussion

The results show us that an auto encoder can be used for 3 dimensional super resolution for facial images, and the performance is much better than simple interpolation. It can be seen that the auto encoder indeed can bring back high frequency components that are lost due to down scaling a high resolution image, or even bring them back in images that never had them due to low quality sensors. However, there are still some drawbacks to using this method, as for now the auto encoder is only trained using images that have the facial features, such as nose, mouth and eyes, in the same place, so it will not be able to reproduce images that have these features in different places, that would require a bigger data set and a deeper auto encoder. The auto encoder is also only trained with one noise level and even though performance for different noise levels is still good, it likely still can be improved using different noise levels for training, however this would require a deeper auto encoder as well, as it was tried using the architecture proposed in this paper but unsuccessfully. It is also likely that overall performance will improve with a deeper auto encoder, as it was shown in this paper that deeper auto encoders give better performance, but due to time limitation no tests were done using deeper architectures. It might also be worth trying out variational auto encoders or coupled deep auto encoders as they have shown promising performance for 2d images in [11] and [10], and are likely to perform well in 3 dimensions as the architecture used in this paper is not different from one that can be used for 2d super resolution.

Another thing is that in the interpolation method used to compare the auto encoder against is very simple and does not have good performance. There are much better methods proposed in [1], [2], [4] to name a few, however no tests were done using these methods so nothing can be said about how the auto encoder compares to these methods.

VI. Conclusion

To conclude, a method has been developed using auto encoders for 3 dimensional super resolution for facial recognition. Regular and fully convolutional architectures with different depths were trained using downscaled images of the FRGCv2 database and tested using other images from the same database. This resulted in the best performance from the deepest fully convolutional auto encoder which was adjusted afterwards and trained for longer using the same images from the FRGCv2 database as before. This architecture was then tested and compared to a simple implementation of interpolation in therms of performance and execution times. The reconstructions were also used for facial recognition from which the FRR and FAR were determined. The two methods were also tested using the FR3DK dataset, which are images from a Microsoft Kinect camera. These test results showed that the auto encoder is much better than the interpolation method, in addition it is still likely that more improvements can be made and even better performance can be achieved.

References

[1] Stefano Berretti, Pietro Pala, and Alberto del Bimbo. Face recognition by super-resolved 3d models from consumer depth cameras. *IEEE Transactions on Information Forensics and Security*, 9(9):1436–1449, 2014.

[2] Minjie Cao, Zhe Liu, Xueting Huang, and Zhuoxuan Shen. Research for face image super-resolution reconstruction based on wavelet transform and srgan. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, volume 5, pages 448–451, 2021.

[3] Yu-Ping Chiu, Jin-Jang Leou, and Han-Hui Hsiao. Super-resolution reconstruction for kinect 3d data. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2712–2715, 2014.

[4] Koji Nishio, Yuta Muraki, Ken-ichi Kobori, and Takayuki Kanaya. Generation of high-resolution and high-precision depth image. In *2018 International Workshop on Advanced Image Technology (IWAIT)*, pages 1–4, 2018.

[5] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, Jin Chang, K. Hoffman, J. Marques, Jaesik Min, and W. Worek. Overview of the face recognition grand challenge. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 947–954 vol. 1, 2005.

[6] F. Schimbinschi, L.R.B Schomaker, and M.A. Wiering. Ensemble methods for robust 3d face recognition using commodity depth sensors. *ECMLPKDD RESEARCH PROCEEDINGS LNCS/LNAI*, 2015.

[7] Manoj Sharma, Santanu Chaudhury, and Brejesh Lall. Deep learning based frameworks for image super-resolution and noise-resilient super-resolution. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 744–751, 2017.

[8] Jan Smisek, Michal Jancosek, and Tomas Pajdla. 3d with kinect. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1154–1160, 2011.

[9] Lieuwe Jan Spreeuwers. Breaking the 99*IET biometrics*, 4(3):169–177, 2015. eemcs-eprint-25873.

[10] Kun Zeng, Jun Yu, Ruxin Wang, Cuihua Li, and Dacheng Tao. Coupled deep autoencoder for single image super-resolution. *IEEE Transactions on Cybernetics*, 47(1):27–37, 2017.

[11] Jinjin Zhang, Di Huang, Yunhong Wang, and Jia Sun. Lock3dface: A large-scale database of low-cost kinect 3d faces. In *2016 International Conference on Biometrics (ICB)*, pages 1–8, 2016.

[12] Qian Zhang, Yuanbo Zhou, Tong Tong, and Qinquan Gao. Image super resolution in real world using variational auto encoder. In *2020 Cross Strait Radio Science Wireless Technology Conference (CSRSWTC)*, volume 1, pages 1–3, 2020.

Appendix

A. Autoencoder Architectures

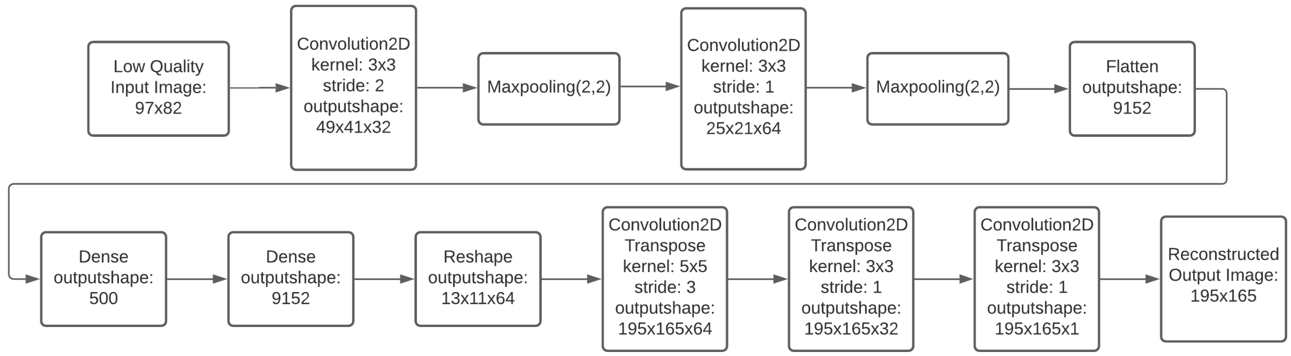


Figure 8: Normal auto encoder with depth = 2

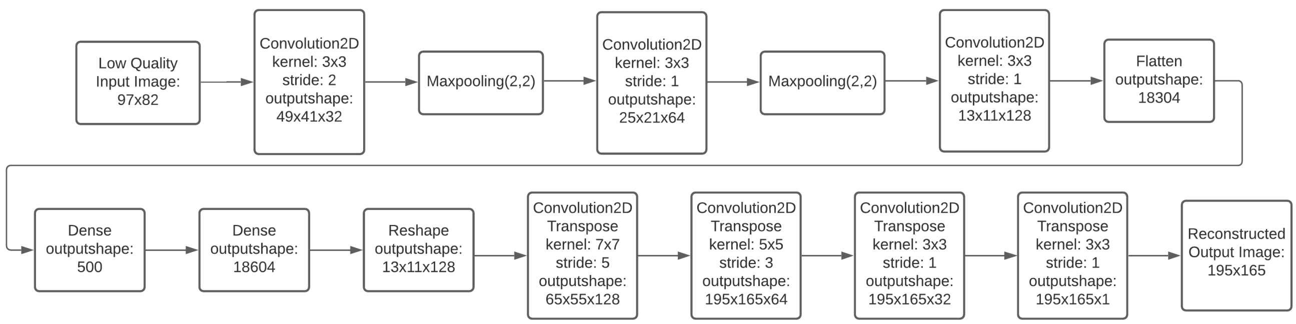


Figure 9: Normal auto encoder with depth = 3

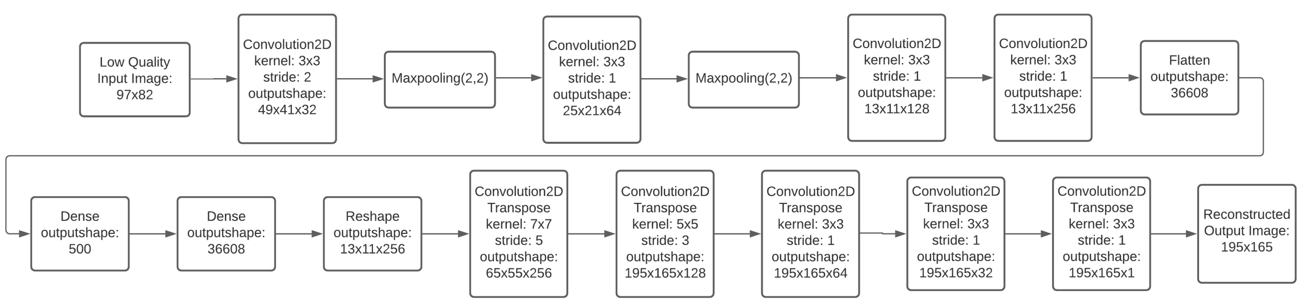


Figure 10: Normal auto encoder with depth = 4

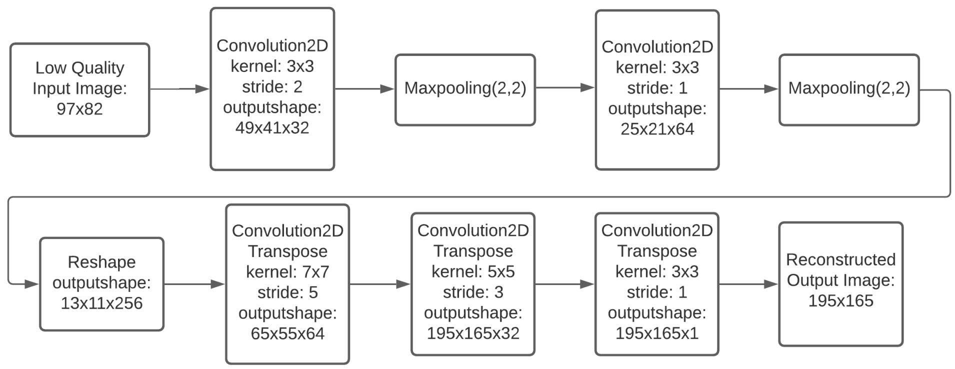


Figure 11: Convolutional auto encoder with depth = 2

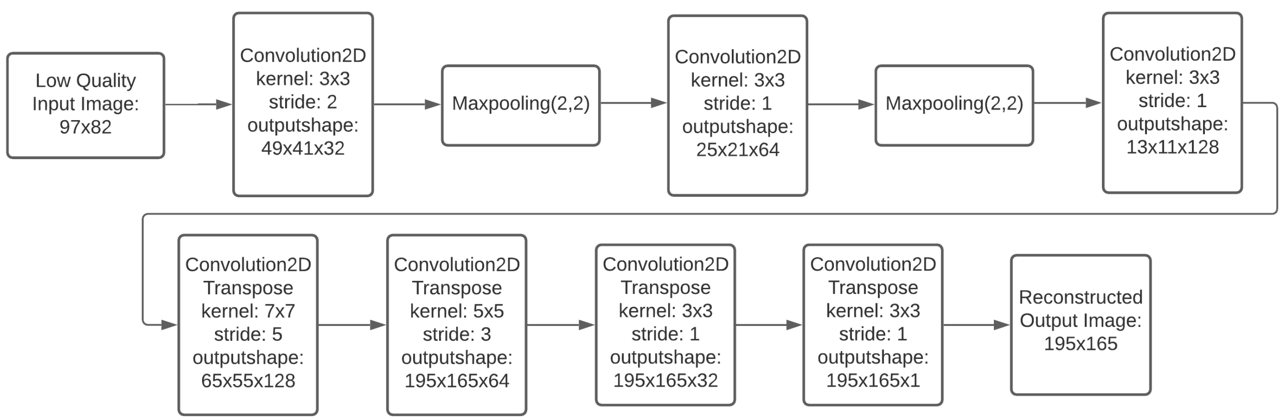


Figure 12: Convolutional auto encoder with depth = 3

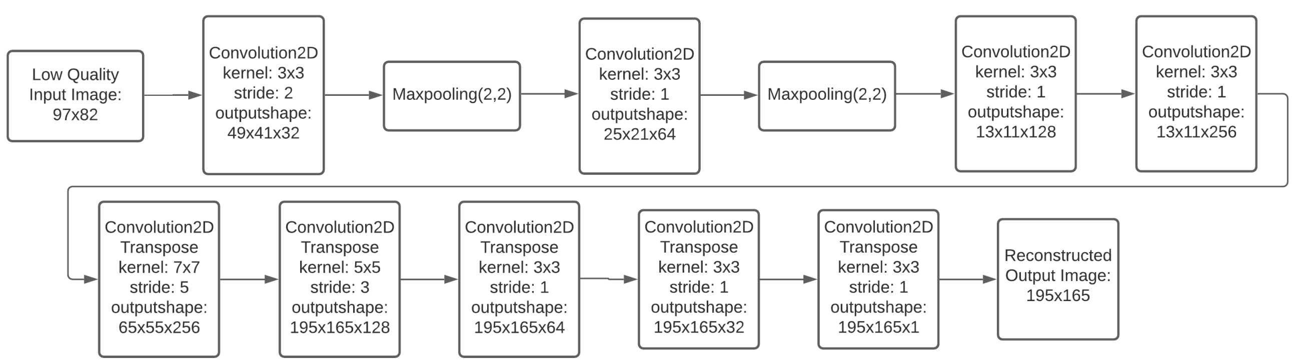


Figure 13: Convolutional auto encoder with depth = 4

B. Original and Low Quality images

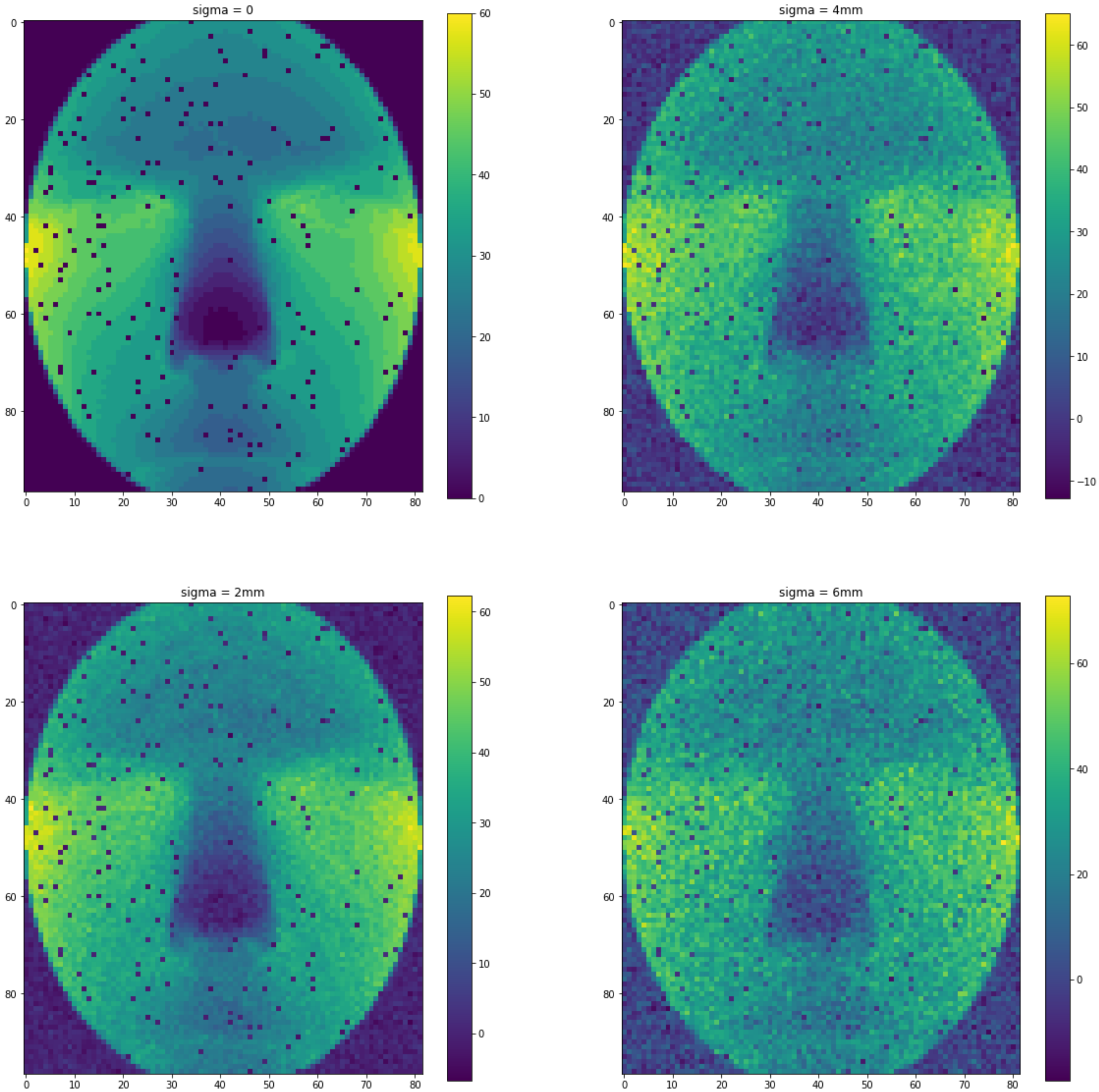


Figure 14: Down scaled images for the different sigmas

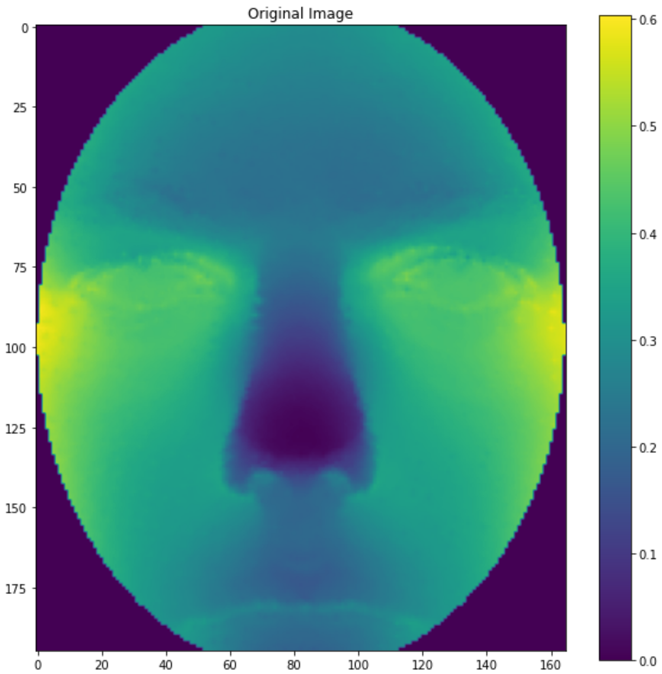


Figure 15: Original image used for the visualizing the reconstructions

C. Reconstruction with different architectures

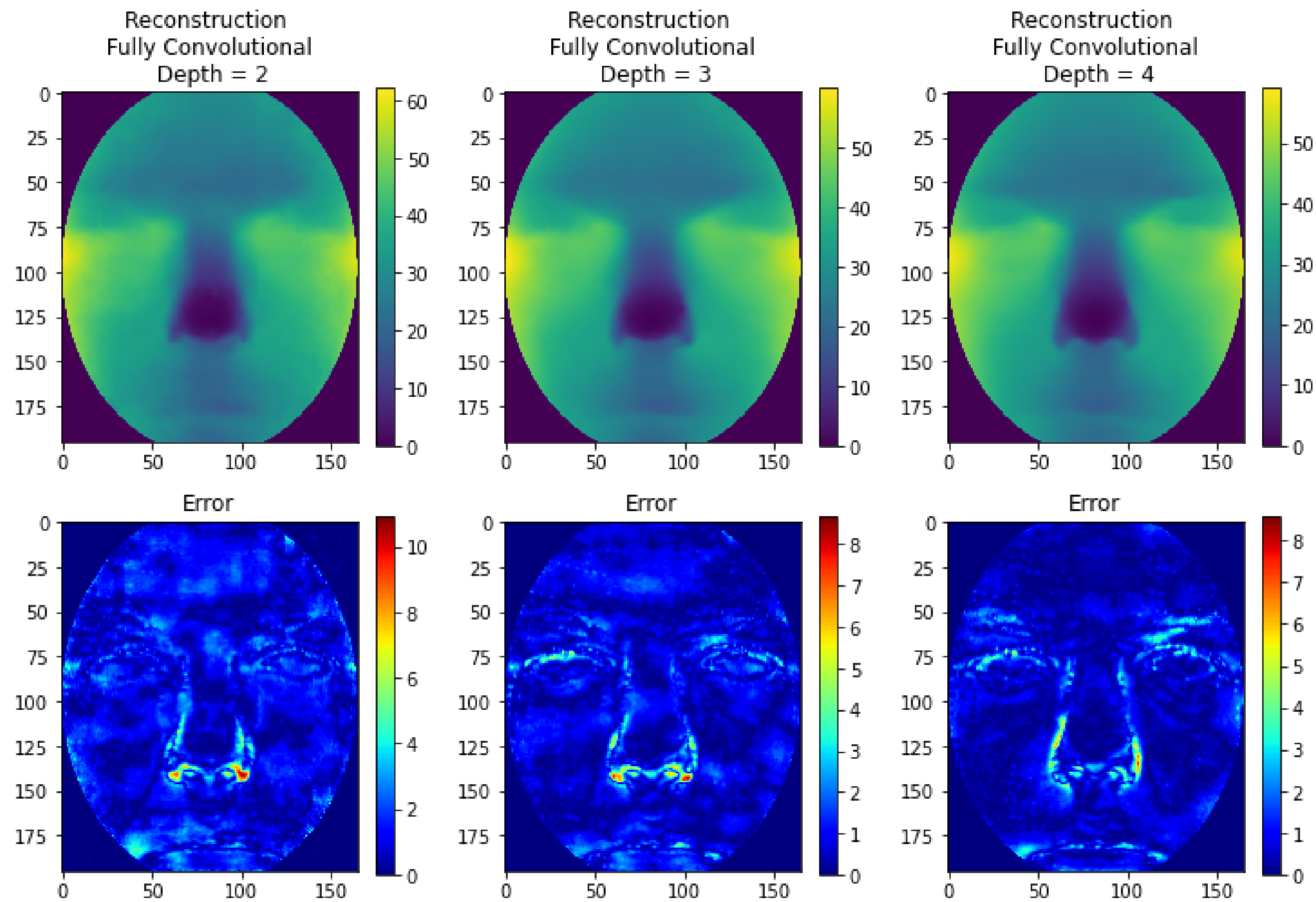


Figure 16: Reconstructions using fully convolutional auto encoders with different depths

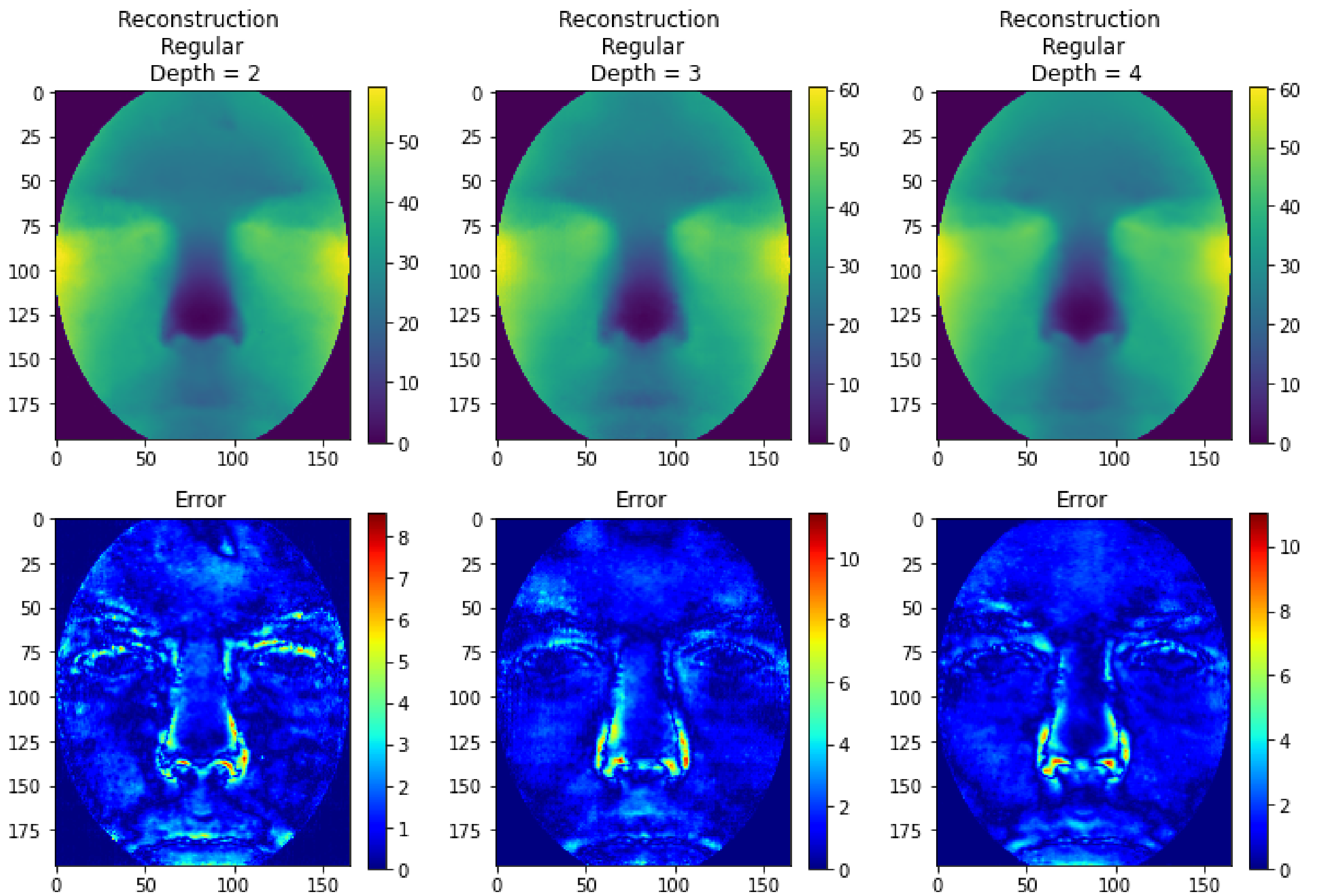


Figure 17: Reconstructions using regular auto encoders with different depths

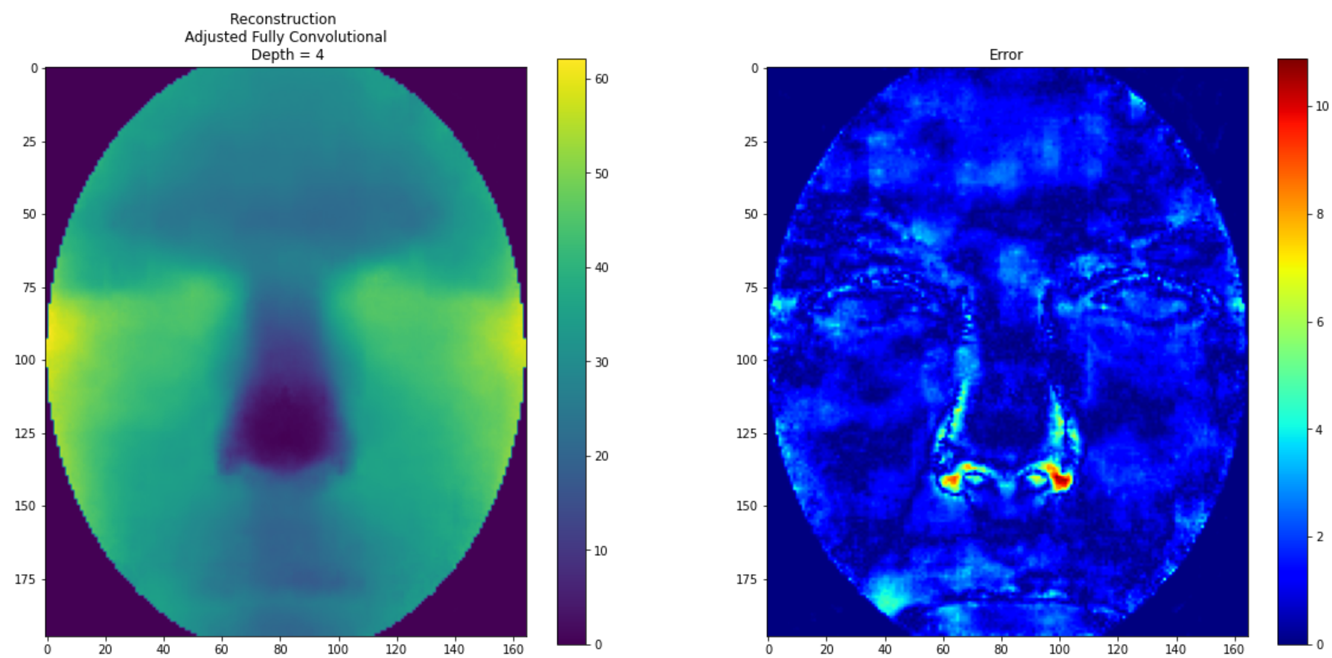


Figure 18: Reconstructions using the adjusted auto encoder architecture

D. Upscaled Images

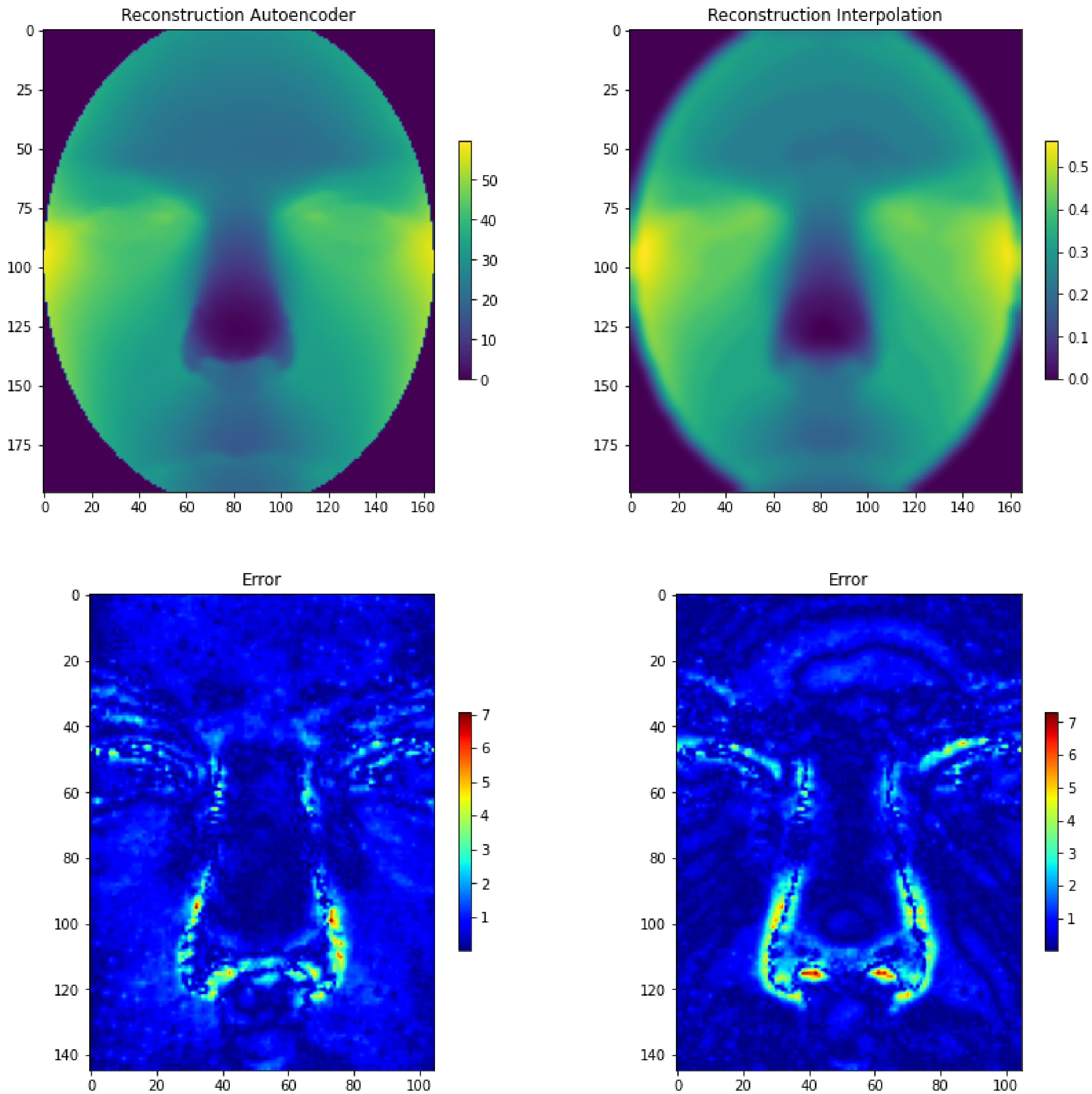


Figure 19: Reconstructions and errors for σ = 0mm

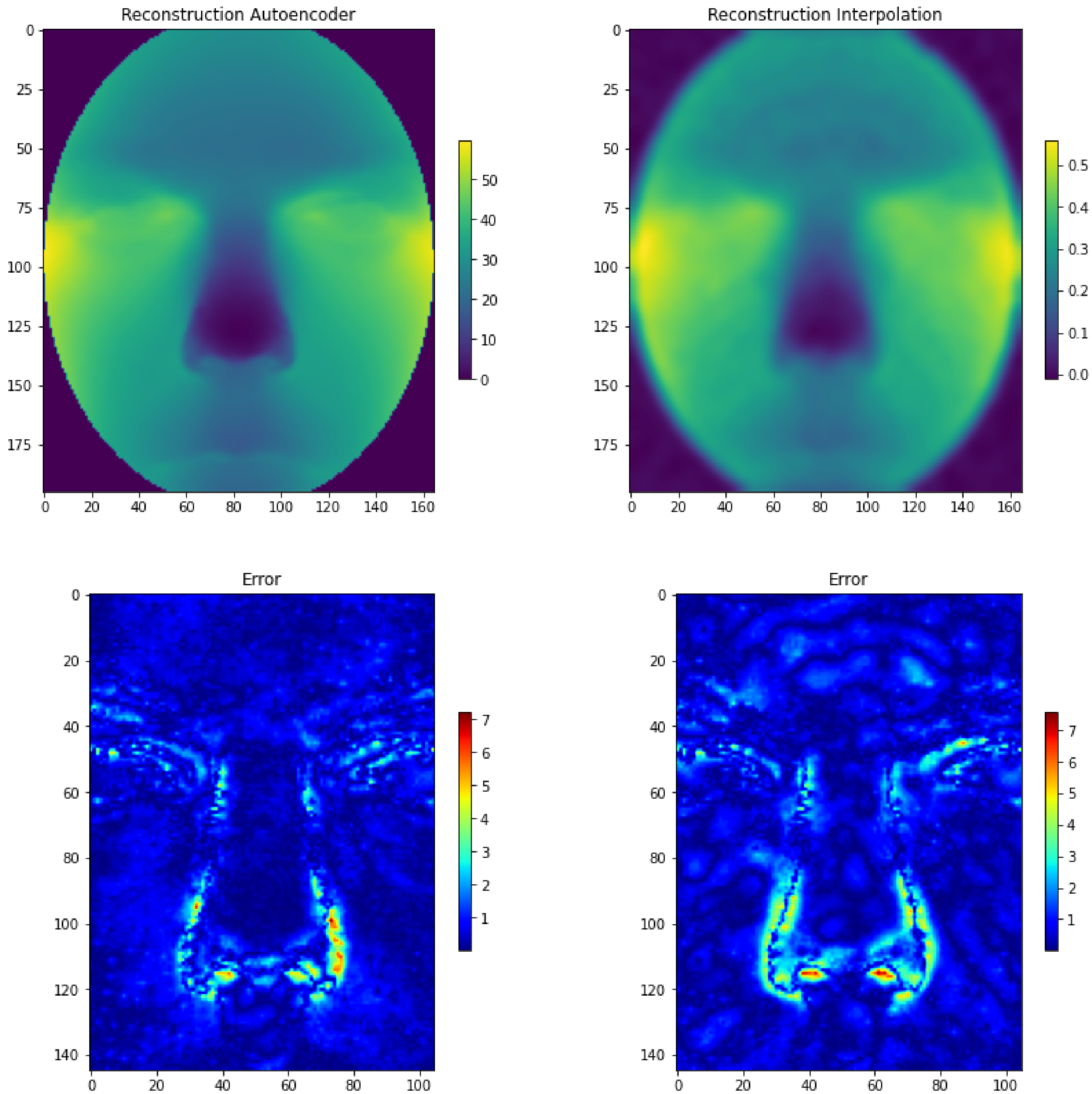


Figure 20: Reconstructions and errors for σ = 2mm

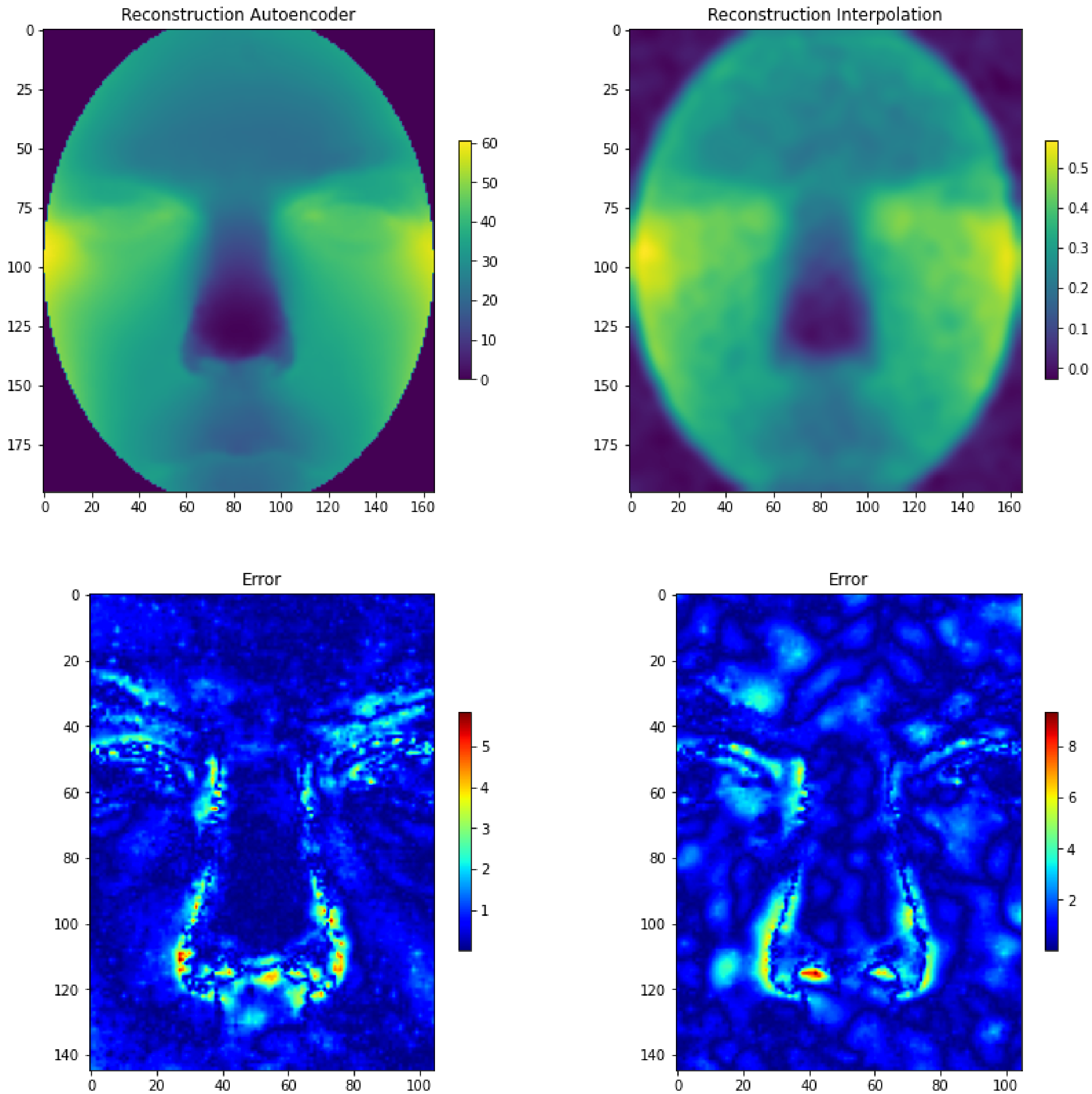


Figure 21: Reconstructions and errors for σ = 4mm

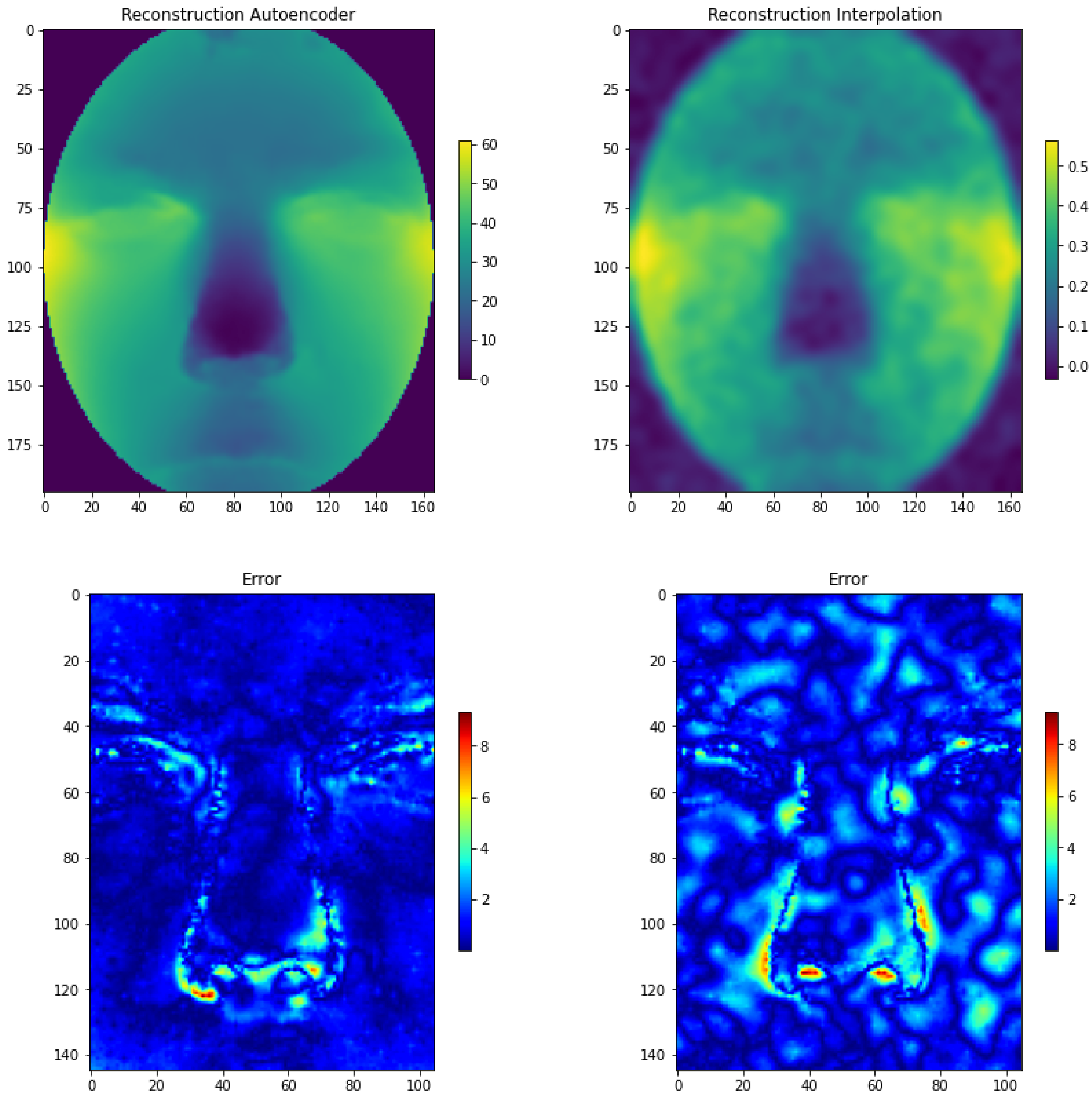


Figure 22: Reconstructions and errors for σ = 6mm

E. False Accept and False Reject

Table V: False accept and False reject seperated by subject

	Auto Encoder vs Original Images		Auto Encoder vs Autoencdoer	
	False Negative	False Positive	False Negative	False Positive
1	25	0	27	3
2	0	0	0	0
3	0	3	5	4
4	8	0	6	0
5	8	0	7	0
6	0	3	0	8
7	0	6	0	7
8	0	0	4	0
9	0	0	0	0
10	0	0	0	7
	Low Quality vs Original Images		Low Quality vs Low Quality	
	False Negative	False Positive	False Negative	False Positive
1	100	0	42	111
2	100	0	34	93
3	100	0	40	112
4	100	0	14	119
5	100	0	12	61
6	99	0	37	24
7	100	0	26	162
8	100	0	18	135
9	100	0	29	38
10	100	0	30	142
	Interpolation vs High Quality		Interpolation vs Interpolation	
	False Negative	False Positive	False Negative	False Positive
1	100	0	32	679
2	100	0	0	579
3	100	0	0	809
4	92	0	0	789
5	90	0	0	758
6	60	0	18	650
7	100	0	0	790
8	99	0	0	822
9	53	0	0	625
10	90	0	0	692

Table VI: False accept and False reject original images

	Original image vs Original image	
	False Negative	False Positive
1	20	1
2	0	0
3	0	0
4	0	0
5	4	0
6	0	0
7	0	2
8	0	0
9	0	0
10	0	1