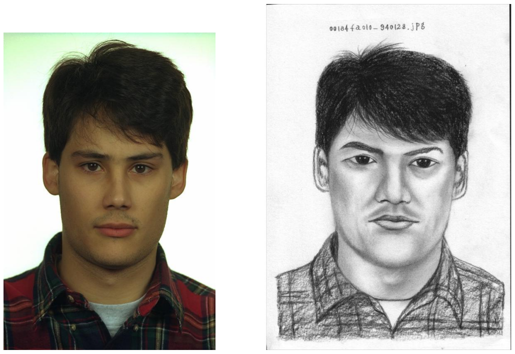
Investigative Study on Generative Models for Face2Sketch and Sketch2Face

Sarthak Jain (36065118)
School of Computing and Communications
Lancaster University
Lancaster, UK
s.jain22@lancaster.ac.uk

***Abstract*—In this report, we present a descriptive analysis of generative models like autoencoders, deep autoencoders, sparse autoencoders, convolutional autoencoders, LSTM, and variational autoencoder for the generation of facial sketches from photos as well as generation of facial photos from sketches. We make use of the ColorFERET data set for training and testing the architectures. We also propose two novel cost functions which enhance the training process to generate better results. The performance of the models when optimizing these cost functions is compared against the performance of the models when using classical loss functions like mean squared error, mean absolute error and binary cross-entropy. The cost functions are also used as metrics for the similarity between the ground truth and the generated output.**

***Index Terms*—computer vision, image generation, generative models, autoencoders.**



(a) Face photograph (b) Face sketch

Fig. 1: An illustration of a face photo and a sketch from the FERET data set

I. Introduction

Generation of facial photos and facial sketches has a plethora of applications in animation generation, enforcement of the law, gaming, as well as other forms of digital entertainment [1]. For instance, many people cherish portraits of themselves or their loved ones and sometimes use them as profile pictures on their social media platforms. However, manually creating such sketches is time consuming, laborious and requires professional artistic skills. Hence, automatic algorithms that can perform this task can be valuable [2].

In terms of enforcement of the law, the photographs of a suspect could prove vital in solving and successfully closing a case. However, in most cases, the photographs are not readily available and the authorities have to rely on the sketches created by the artists based on the description given by eyewitnesses. Thus, an automatic model for the generation of facial photographs can prove very useful [3], [4], [5], [6], [7].

Face photograph-sketch generation is a task involving computer vision, heterogeneous image transformation [8], image-to-image translation [9] and image style transfer [10]. It can be used to solve these real-life problems. The core challenge of the generation of face photo-sketch is producing realistic images that surpass the obstacles created by shape, texture and colour [11].

In this report, we analyze the working of various generative models for both the generation of facial photographs from sketches and facial sketches from photographs A brief description of the methods used is given in the preliminary on methodologies. We also consider the variations in their architectures to compare the effects on the performance of the models which are discussed at length in the Experimental results section. We propose two new cost functions to optimize the image generation problem that helps in generating considerably better results for simpler models.

We utilize the FERET data set for our analysis, which is a large database of facial images, divided into development and sequestered portions [12], an example of the same is shown in Fig. 1. For Image to sketch generation, we transform the images into grayscale images as colour is not an important feature. This also helps in optimizing the training time. For the sketch to image generation, the facial photos used are RGB images with 3 respective colour channels. As a baseline model, we utilized the basic autoencoder. All the images are resized to have 128 × 128 pixels. The findings of the analysis are discussed in depth in the discussions sections, followed by the conclusion and future scope.

II. Literature Review

There are two broad approaches for the facial photo-sketch generation, namely the exemplar-based methods and the deep

Sincere thanks to Dr. Richard Jiang, Sayantee Dutta and Vaibhav Khandelwal for their support and guidance.

learning-based methods [11]. The exemplar-based methods aim for the reconstruction of target images by using sample images from the training data set to learn patterns based on the patch-to-patch similarity between the input image and the target image. The same is illustrated in Fig. 2. These patterns are then used to make predictions for images from the test data set. However, matching the patches between the images is very time-intensive. Exemplar-based methods can be sub-categorized into subspace learning-based approaches [7], sparse representation-based approaches [13], and Bayesian inference-based approaches [14].

[15], an exemplar-based approach for face sketch synthesis was leveraged by assuming a linear mapping between the face photos and face sketches. The eigenvectors of the sample sketch images were then used for generating the corresponding face photos. However, the assumption of linear mapping between the photos and sketches does not always hold, especially when considering the hair regions of the images.

In [7], the authors proposed a nonlinear approach of considering the images as patches that overlap each other. Further, Local Linear Embedding(LLE) is used to reconstruct the target image where the average is considered for the areas where the patches of the image in the vicinity to each other overlap. This approach resulted in a block effect. Following a similar approach, [16] utilized Markov Random Field (MRF) to map the distribution between the image patches. A major problem with this approach was the new patches could not be generated and resulting in a sub-optimal solution.

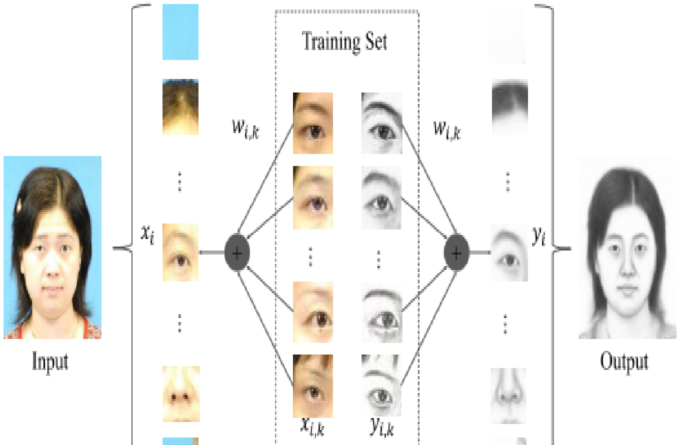


Fig. 2: Exemplar-based approach of facial photo-sketch generation [2]

Deep-learning based methods make use of deep learning-based architectures to transform the image generation problem into a mathematical optimization problem. This is illustrated in Fig. 3. These models evolved gradually and recently grew in popularity with the advances in computing capabilities of the modern-day systems, as well as the developments in non-linear models like Convolutional Neural Networks(CNNs) [17], which formulate a non-linear mapping between the sketches and photographs.

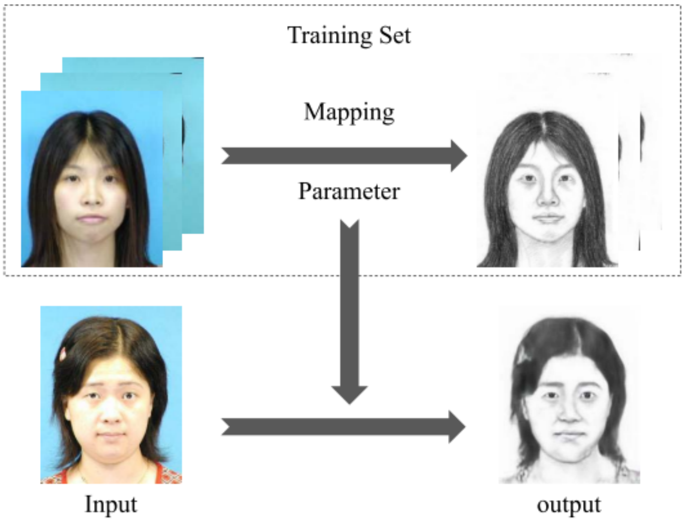


Fig. 3: Deep learning-based approach of facial photo-sketch generation [2]

In [18] an end-to-end Fully Convolutional Neural Network for generating sketches from photographs is used by the authors. The architecture aimed to find a non-linear mapping between the images and sketches. The results were affected by the lack of depth in the architecture and the use of the Mean Squared Error(MSE) loss function.

Generative Adversarial Networks(GANs) have recently gained much popularity for image generation tasks. Both [9] and [19] utilize conditional and cyclic variations in GANs, respectively to perform image-to-image translation tasks. Thus, these models can also be used for the face photo-sketch generation tasks.

In [20], a multi-scale discriminator to provide adversarial supervision on different image resolutions is proposed by the authors. In [21], a semi-supervised learning algorithm is proposed for data augmentation to the pre-trained data.

All these approaches helped in achieving significant progress in face photo-sketch generation, but all of these have certain shortcomings when used in real-life scenarios.

III. Preliminary on Methodologies

A. *Loss functions and Performance Metrics*

The models that are analyzed in this report are generative models where the image generation problem is modelled to an optimization problem, where a cost function is optimized, and the weights of the model are adjusted in a way that the output generated matches the target image as closely as possible. Hence, the loss/cost function used by a model becomes very important. In this report, we propose two novel loss functions for image synthesis and compare their performance against some standard loss functions all of which are described below:

- **mean squared error(MSE) [22].** It is generally used to measure the quality of the image generated in image compression problems. It is derived by the eq. 1 where $\mathbb{E}\{.\}$ denotes the expectation operation. $\hat{f}$ denotes the reconstructed image and f denotes the original image.

$$MSE(\hat{f}) = \mathbb{E}\{\|f - \hat{f}\|^2\} \tag{1}$$

- **mean absolute error(MAE) [23].** Like MSE, MAE is generally utilized for calculating errors in image compression problems and poses similar issues for face photo-sketch synthesis. The advantage of using MAE is in assessing the model performance. It is given by eq. (2).

$$MAE(\hat{f}) = \mathbb{E}\{\|f - \hat{f}\|\} \tag{2}$$

- **Binary Cross-entropy(BCE) [24].** When the pixel values of an image are scaled to have values between 0 and 1, pixel values can be thought of as the probability of the pixel being illuminated. In such a situation, binary-cross entropy can be used as the loss function with images. It is given by eq. (3), where n is total number of pixels in the images, y_i and $\hat{y}_i$ denote the i^{th} pixel values for original and reconstructed images, respectively.

$$BCE = -\frac{1}{n}\sum_{i=1}^{n} y_i \times \log \hat{y}_i + (1 - y_i) \times \log(1 - \hat{y}_i) \tag{3}$$

In an image compression problem, a lower quality image is generated from an existing image. However, face photo-sketch generation is a heterogeneous image transformation problem, where the input passed, and the output derived don't match exactly. Thus, the usage of the above-mentioned cost functions in such problems is not ideal due to the pixel-by-pixel nature of error calculation. Instead, usage of loss functions that consider the distribution of pixel intensity values in an area in the image. Therefore, the proposed loss functions consider both pixel-to-pixel similarities between the generated and target images as well as the distribution of intensity values. The first new loss function(newLoss1) is given by eq. (4),

$$newLoss1 = (\frac{100}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2) + (0.0001\sum_{i=1}^{n} y_i \log(\frac{y_i}{\hat{y}_i})) \tag{4}$$

whereas the second new loss function(newLoss2) is given by eq. (5).

$$\begin{aligned} newLoss2 = (\frac{100}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2) + \\ (0.00005\sum_{i=1}^{n} y_i \log(\frac{y_i}{\hat{y}_i}) + \sum_{i=1}^{n} \hat{y}_i \log(\frac{\hat{y}_i}{y_i})) \end{aligned} \tag{5}$$

The second term in both the loss functions considers the distribution of pixel intensity values while the first term calculates the error in the pixel-to-pixel structure of the images. A weighted summation of both the terms is used in the models to generate the results. It is to be noted that while the newLoss1 is not symmetric, newLoss2 is designed to be symmetric.

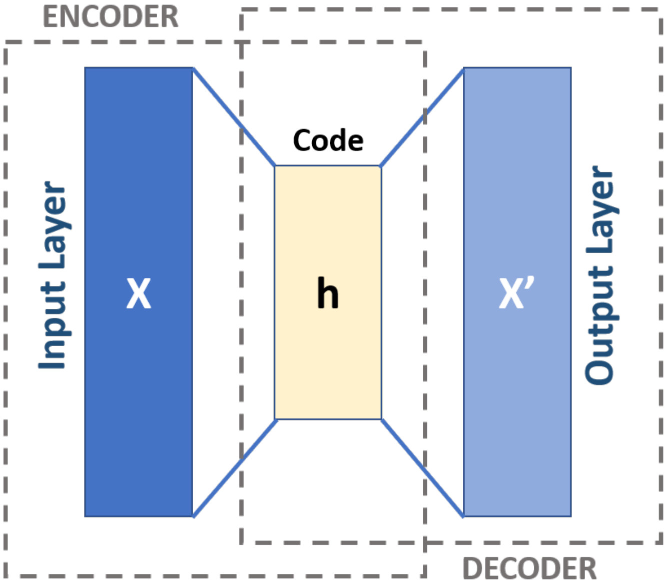


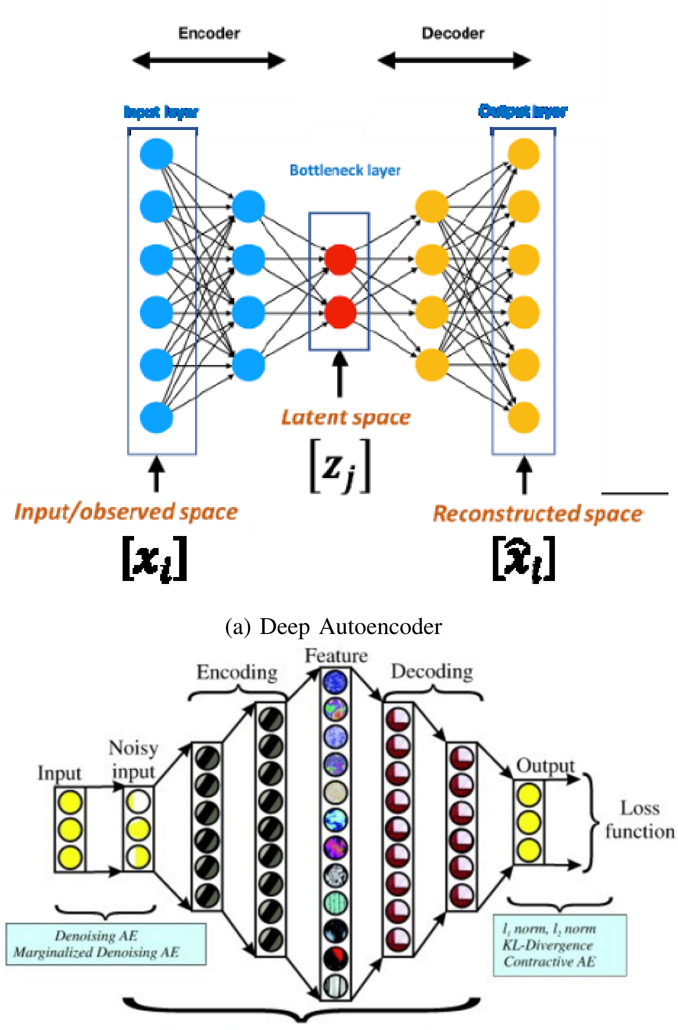
Fig. 4: Basic Autoencoder schema

B. Baseline Model - normal Autoencoder

The basic autoencoder [25] is a simple artificial neural network that consists of 3 sections: an encoder section, the latent space and the decoder section which is also shown in fig. 4. The autoencoders are predominantly used for image compression, dimensionality reduction and denoising images. The task of the encoder is to take an input and learn the underlying patterns in the input in an encoded format, which can later be used to generate an output. The output is generally a reconstructed input or can be a denoised version of the input. We use the basic autoencoder as our baseline model.

C. Deep Autoencoders and Deep Sparse Autoencoders

The deep autoencoders are simply a variation of the basic autoencoder with more hidden layers in the encoder as well as the decoder part. As the model has more layers it can learn more underlying patterns in the input. Thus, deep autoencoders are expected to generate better results than basic autoencoders. We consider two different architectures of deep autoencoder where the number of neurons in every subsequent layer of encoder decrease while they increase in the decoder and vice-versa which is sometimes known as deep sparse autoencoder. There is a variation in the size of the latent space in both models. Sparse autoencoders have a bigger latent space which can allow them to learn more patterns from the input The features stored in the latent space are sparse features. However, they are not that popular in use. Both the architectures contain a total of 10 layers with the number of neurons in the hidden layers varying between 1024 and 64. The training data is reshaped into a 2D to be passed as an input to the models.



(a) Deep Autoencoder

(b) Deep sparse Autoencoder

Fig. 5: An illustration of deep autoencoder and deep sparse autoencoder

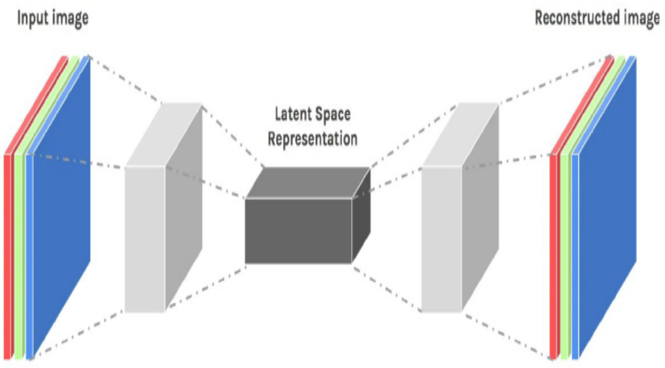


Fig. 6: Convolutional Autoencoder

D. Convolutional Autoencoders

Convolutional autoencoders are one of the most popular types of autoencoders in computer vision tasks. They have a similar architecture to a normal autoencoder. However, the inputs and the neurons in adjacent layers of the model are connected through convolutions. The convolutions can be thought of as kernel functions that traverse over a portion of the input and generate a feature map. These feature maps form the input to the subsequent layers. The encoder section of the network makes use of the pooling operations to reduce the size of the input to the next layer until the encoded patterns are learned and stored in the latent space. To generate the final output, up-sampling operations are performed between the layers of the network.

The above-mentioned models are known to generate blurry images and the convolutional networks are usually expected to perform better as they are known to have more expressive power than the normal deep architectures. With every subsequent layer, the model learns more complex patterns in the input which is why they are widely utilized in several use-cases.

E. Variational Autoencoders

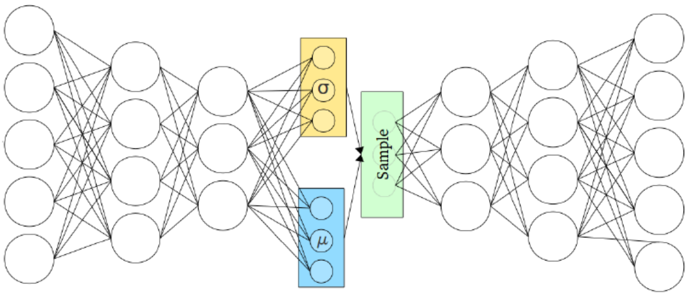


Fig. 7: Variational Autoencoder

We make use of variational autoencoders for our use case as they can learn the probability distribution of the training data set, which is a task not performed by the models discussed above. The encoder of the variational autoencoder makes use of two latent spaces in parallel which are used to model the Probability distribution of the input data set. While one latent space learns the mean of the distribution the other learn the variance. The decoder then generates samples from the latent space which is then used to generate the final output. The architecture of the variational autoencoder is illustrated in fig. 7.

F. Long Short-Term Memory (LSTM)

LSTMs [26] are a variation of Recurrent neural networks(RNNs). They are networks that can persist information this is possible due to the existence of loops in the architecture which allow the information from one step of the network to the next. However, the RNNs suffer from short-term memory. If the input given is large, RNNs find it difficult to propagate the information from one step to another. LSTMs solve this problem of the RNNs by using the mechanism called gate which controls the flow of information across the network. The gates can learn which data is relevant and should be propagated to the next step and which data should be discarded. The architecture of the LSTM is illustrated by fig. 8.

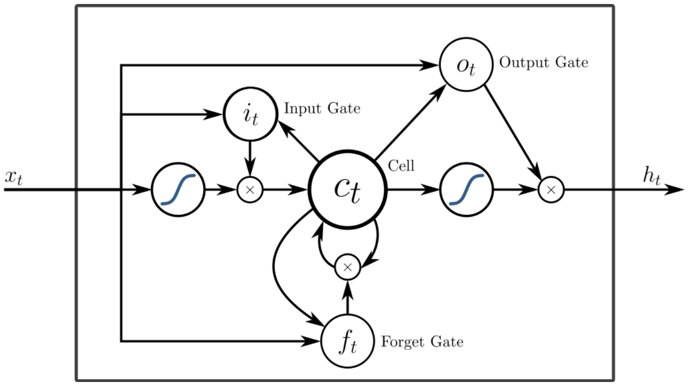


Fig. 8: LSTM

IV. Experimental Results

A. Experimental Setup

a) Running environment and the data directory: As image generation tasks that are performed in our analysis are computationally intensive, the python code is hosted on google colaboratory, a product from Google Research. It provides users with a platform to execute their python code in a jupyter notebook style and proves ideal for machine learning and data analysis. It provides users with a python environment that is preloaded with all the useful libraries and thus, requires no setup. Moreover, free GPUs are provided to the users to perform tasks that require high computation [27].

We use Tensorflow, Keras libraries to implement the architectures and define the custom loss functions, scikit-image and open cv libraries to pre-process the images and for data augmentation, os and glob libraries are used to access and read the required files while NumPy and Matplotlib libraries are used for image visualization.

The dataset is stored on google drive. The facial photos are stored in a folder named photos while the facial sketches are stored in a folder named sketches.

the loss functions used to measure the similarity between the target and generated images are discussed in the Preliminary on Methodologies section.

b) Pre-processing: We use the FERET data set for the analysis in this report. The data set consists of 846 facial photos and the corresponding sketches made manually by the artist. The images contain a mixture of both coloured and grayscale images. The images are of different sizes as well. Hence, all the images are resized to contain 128×128 pixels. For face to sketch generation, all the images are transformed into grayscale images as colour. Essentially, colour is eliminated in the feature selection exercise to optimise the model training time. For sketch to face image generation the faces passed as target values are coloured images as the colour is one of the features that the model needs to learn. Before passing the images to the model, the pixel intensities of each image are normalized by dividing them by 255. This helps in a smoother learning process.

To avoid underfitting in the models, the pre-trained data is augmented by adding to the data set the horizontally flipped images. The heuristics of the image and the facial features don't change in this transformation. Horizontal flipping of an image is illustrated in Fig. 9.

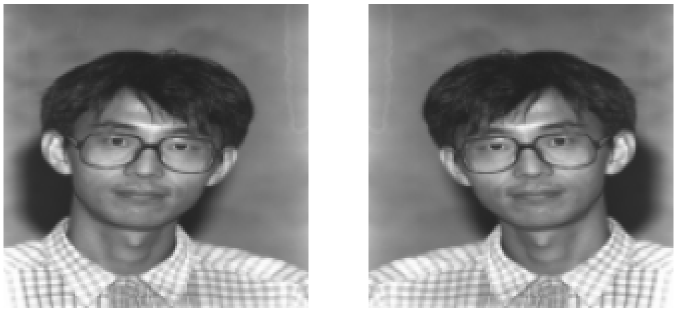


Fig. 9: Illustration of original image (left) and the horizontally flipped image (right)

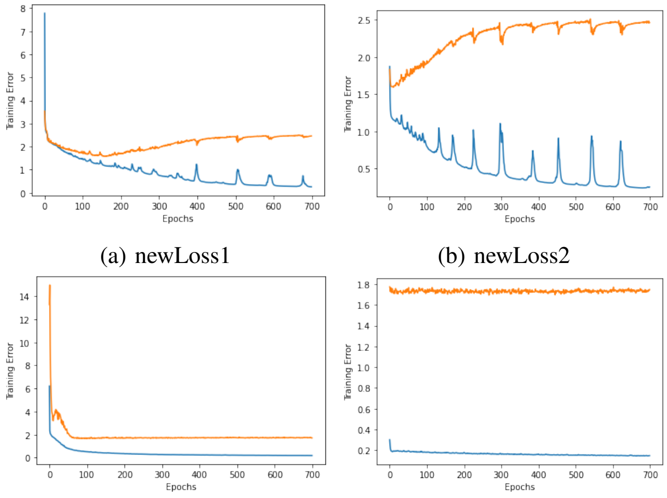
B. Training of the Methods

Each method discussed in this report is trained on a different number of epochs depending as the time taken by the model to complete the training processes is different. To generate better results the models are required to be trained over many epochs. Thus, a tradeoff between time, resources and training had to be made. Fig. 10 shows some of the training curves generated. It is to be noted that the blue curves in each graph show the variations in training losses over the epochs while orange curves showcase the variations in validation loss with the number of epochs.

When generating facial sketches from photographs the training process is faster except for the LSTM network which displays a much slower conversion rate. When using newLoss2 with the LSTM network, the changes in the loss value are so small that they don't change the weights of the model at all. Thus, the graph for the learning curve, in this case, is not generated. The learning curves corresponding to the LSTM network are illustrated in Fig. 11.

C. Test Results for Comparison

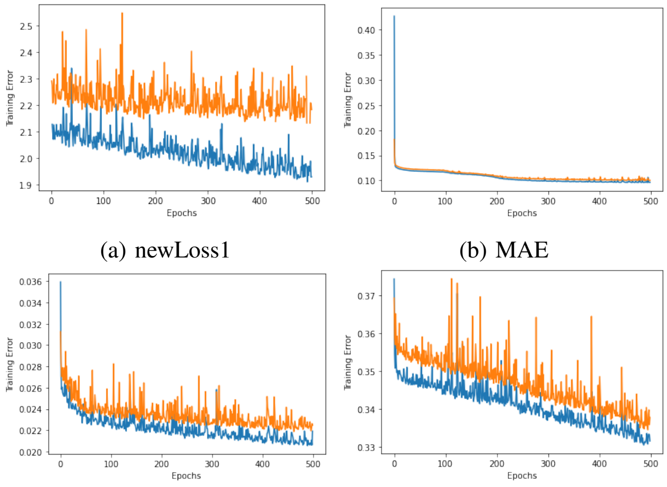
The data set was divided into training and validation data. The prediction results for generating sketches from faces are illustrated in fig. 12 and the results of generating the faces from sketches are illustrated in fig. 13. It is observed that for generating the facial sketches the best results are obtained with deep neural networks optimising the proposed cost function newLoss1 and newLoss2. For generating facial photos from sketches the best results are convolutional neural networks with sparse architecture.



(a) newLoss1 (b) newLoss2

(c) newLoss1 (d) newLoss2

Fig. 10: Training curves for DAE and CAE for the given loss functions when generating sketch from face



(a) newLoss1 (b) MAE

(c) MSE (d) BSE

Fig. 11: Training curves for the loss functions when generating sketch from face with LSTM

When generating facial images from sketches, we ran into a problem of GPU resource exhaustion for normal convolutional networks and LSTM networks. The resources in the Colab environment are not guaranteed to be readily available as well.

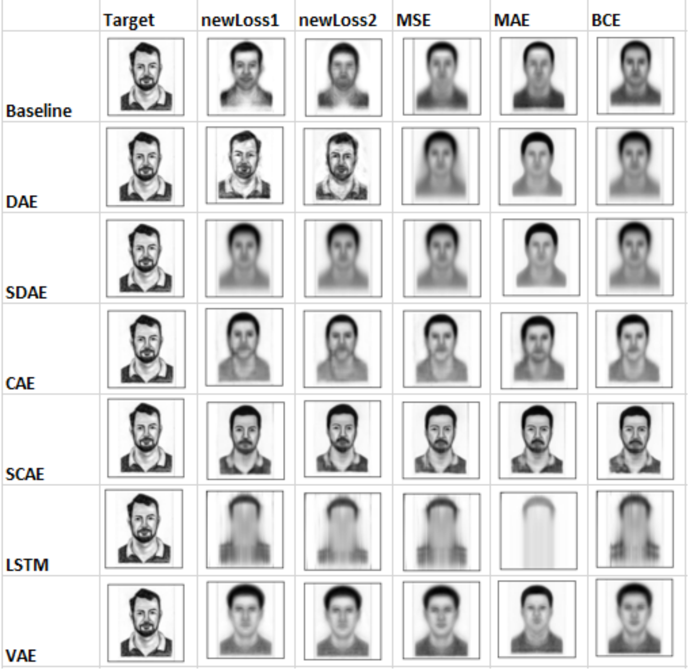


Fig. 12: Test Results for generating facial sketches from images

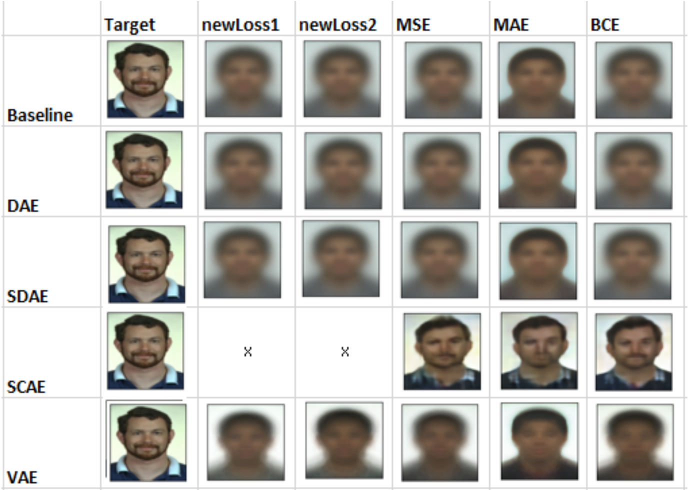


Fig. 13: Test Results for generating facial images from sketches

V. Discussion

The novel loss functions newLoss1 and newLoss2 help the simpler models (AE and DAE) to perform significantly better for face2sketch generation. Thus, using these the results obtained are much better than the standard loss functions. Further, all the models analyzed showcase better results when using the proposed loss function for this task. However, the loss functions are defined in such a way that they work efficiently with single-channel images while they don't fully support RGB images. The function generates an output where the input coloured images are flattened before passing to the models but even it is observed that the models fail to converge. However, when inputs are passed without flattening these functions throw an error. The loss functions can be enhanced to handle coloured image inputs.

Better images are generated with normal DAE when compared to Deep sparse AE but vice versa is true for CAEs which perform significantly better with a sparse architecture.

For face2sketch generation using BCE loss functions provides better results when compared to MSE and MAE. While for sketch2face generation MAE outperforms all the loss functions analyzed.

Although better results are obtained for DAE when using newLoss1 and newLoss2, the overall performance of CAE is much better across all the cost functions. The results obtained show that the variation in loss functions have a relatively smaller effect on the performance of the CAE making them more robust. This makes CAE the preferred choice for such computer vision tasks. Given better and more training time CAEs might outperform the best results of the DAE.

Training VAE takes significantly less time. However, the number of epochs required for the model to converge is significantly higher. Therefore, VAE might generate better results with better resources and more training time.

VI. Conclusion

In this report, we analyzed the performance of various generative models for generating facial sketches from photos as well as facial photos from sketches using the FERET data set. The analysis involved variations in the architecture of the models as well as the cost functions used. We proposed two novel cost functions which perform significantly well with grayscale images for the generation of facial sketches from photos when compared to other standard cost functions. We discussed preprocessing of the data set, the various methods used to implement the generative models and concluded that the best results are obtained for simpler DAE using the novel cost functions for face2sketch generation. However, the CAE is observed to be more robust to variations in cost functions and thus, the overall performance of the model is considerably better than the other models in consideration for both the image processing tasks. Further, more models like GANs can be analyzed for the image generation tasks. Also, the proposed loss functions can be enhanced to support coloured images as well.

References

[1] N. Wang, D. Tao, X. Gao, X. Li, and J. Li, “A comprehensive survey to face hallucination,” *International journal of computer vision*, vol. 106, no. 1, pp. 9–30, 2014.

[2] M. Zhu, J. Li, N. Wang, and X. Gao, “A deep collaborative framework for face photo–sketch synthesis,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 10, pp. 3096–3108, 2019.

[3] W. Konen, “Comparing facial line drawings with gray-level images: a case study on phantomas,” in *International Conference on Artificial Neural Networks*. Springer, 1996, pp. 727–734.

[4] X. Gao, J. Zhong, J. Li, and C. Tian, “Face sketch synthesis algorithm based on e-hmm and selective ensemble,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 4, pp. 487–496, 2008.

[5] R. G. Uhl and N. da Vitoria Lobo, “A framework for recognizing a facial image from a police sketch,” in *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 1996, pp. 586–593.

[6] X. Tang and X. Wang, “Face sketch recognition,” *IEEE Transactions on Circuits and Systems for video Technology*, vol. 14, no. 1, pp. 50–57, 2004.

[7] Q. Liu, X. Tang, H. Jin, H. Lu, and S. Ma, “A nonlinear approach for face sketch synthesis and recognition,” in *2005 IEEE Computer Society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. IEEE, 2005, pp. 1005–1010.

[8] N. Wang, J. Li, D. Tao, X. Li, and X. Gao, “Heterogeneous image transformation,” *Pattern Recognition Letters*, vol. 34, no. 1, pp. 77–84, 2013.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[10] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.

[11] M. Zhu, C. Liang, N. Wang, X. Wang, Z. Li, and X. Gao, “A sketch-transformer network for face photo-sketch synthesis,” in *International Joint Conference on Artificial Intelligence*, 2021.

[12] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, “The feret database and evaluation procedure for face-recognition algorithms,” *Image and vision computing*, vol. 16, no. 5, pp. 295–306, 1998.

[13] L. Chang, M. Zhou, Y. Han, and X. Deng, “Face sketch synthesis via sparse representation,” in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010, pp. 2146–2149.

[14] M. Zhu, N. Wang, X. Gao, and J. Li, “Deep graphical feature learning for face sketch synthesis,” in *Proceedings of the 26th international joint conference on artificial intelligence*, 2017, pp. 3574–3580.

[15] X. Tang and X. Wang, “Face sketch synthesis and recognition,” in *Proceedings Ninth IEEE International Conference on Computer Vision*. IEEE, 2003, pp. 687–694.

[16] X. Wang and X. Tang, “Face photo-sketch synthesis and recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 11, pp. 1955–1967, 2008.

[17] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 international conference on engineering and technology (ICET)*. Ieee, 2017, pp. 1–6.

[18] L. Zhang, L. Lin, X. Wu, S. Ding, and L. Zhang, “End-to-end photo-sketch generation via fully convolutional representation learning,” in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, 2015, pp. 627–634.

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

[20] L. Wang, V. Sindagi, and V. Patel, “High-quality facial photo-sketch synthesis using multi-adversarial networks,” in *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 2018, pp. 83–90.

[21] C. Chen, W. Liu, X. Tan, and K.-Y. K. Wong, “Semi-supervised learning for face sketch synthesis in the wild,” in *Asian Conference on Computer Vision*. Springer, 2018, pp. 216–231.

[22] N. P. Galatsanos, M. N. Wernick, A. K. Katsaggelos, and R. Molina, “3.7 - multichannel image recovery,” in *Handbook of Image and Video Processing (Second Edition)*, second edition ed., ser. Communications, Networking and Multimedia, A. BOVIK, Ed. Burlington: Academic Press, 2005, pp. 203–217. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780121197926500760

[23] J. H. Lin, T. M. Sellke, and E. J. Coyle, “Adaptive stack filtering under the mean absolute error criterion,” in *Nonlinear Image Processing*, vol. 1247. International Society for Optics and Photonics, 1990, pp. 182–193.

[24] A. Creswell, K. Arulkumaran, and A. A. Bharath, “On denoising autoencoders trained to minimise binary cross-entropy,” *arXiv preprint arXiv:1708.08487*, 2017.

[25] P. Baldi, “Autoencoders, unsupervised learning, and deep architectures,” in *Proceedings of ICML workshop on unsupervised and transfer learning*. JMLR Workshop and Conference Proceedings, 2012, pp. 37–49.

[26] C. Olah, “Understanding lstm networks,” Aug 2015. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[27] “Google colaboratory faq.” [Online]. Available: https://research.google.com/colaboratory/faq.html