Deep Painter: Painter Classification using Deep Convolutional Autoencoders

E4040.2021Fall.MSKD.report.km3714.mr4136.sc4921
Kiran M Sreenivasa km3714, Manisha Rajkumar mr4136, Sunjana Ramana Chintala sc4921
Columbia University

Abstract

The seed paper illustrates the novel architecture of Deep Convolutional Autoencoders and effectively demonstrates the usefulness of this framework to train and classify a dataset of paintings. This paper highlights the distinction of Deep Convolutional Autoencoder by presenting an analysis along with commonly used techniques that rely on image processing and feature extraction. Our study is an extension of the seed paper. Our project focuses on tackling the 3-painter classification task by training the proposed Deep Convolutional Autoencoder on numerous images and utilizing it to initialize a Supervised CNN in classification phases of 3-painter and 5-painter problems. Further, our project outlines a comparative analysis of the approach with Deep Learning Classifiers like VGG16, Xception, MobileNet and ResNet50 models.

1. Introduction

Deep Convolutional Autoencoders or Convolutional Autoencoders(CAEs) are Convolutional Neural Networks in which for every convoluted layer and max-pooling layer, a corresponding deconvoluted layer and unpooling layer exists[2,3,4]. They are essentially a combination of CNNs and Autoencoders. CAEs are used to initialize a supervised CNN to resemble the training data and perform classification tasks.
In recent years, painting authentication has gained significant importance, given the increase of Art forgery. The seed paper[1] tackles the task of Automatic Classification of paintings (a closely related problem) using Deep Convolutional Autoencoders.
The authors explore the 3-painter classification problem by constructing the CAE architecture to operate on the raw pixel level without the necessity of any preprocessing or manual feature extraction.
One challenge in most classification tasks is feature detection. In this paper, the CAE framework is designed, primarily, with convolutional layers and max-pooling layers to find meaningful features like colour and composition that closely match real-world images. Another challenge in classification tasks is to get rid of noisy data. The authors of this paper make use of concepts like "periodic multiplication of learning rates" to address this difficulty and denoise the data. While training the CAE, the authors use decoder components. The Classification task is performed with a fully trained CAE initialized on a Supervised CNN. Finally, the authors also underscore the merit of CAEs by comparing the accuracy with previous classification methods.

2. Summary of the Original Paper

2.1 Methodology of the Original Paper

In the paper[1], the authors point out that Convolutional Neural Networks have been receiving a great amount of attention in the past decade because of their ability to perform better than traditional image processing techniques. A typical CNN's architecture consists of a convolutional layer, a max-pooling layer and a softmax output layer for performing classification tasks. In the paper, the authors venture to classify the paintings of three artists. Since the number of samples presented for each painter are few in number, the authors decide to use an unsupervised pre-training technique which is an Autoencoder. An autoencoder makes use of standard backpropagation for unsupervised training of the model.

The authors' name this combination of CNNs and Autoencoder as a Convolutional Autoencoder. A Convolutional Autoencoder makes use of both the principles to effectively classify images with fewer samples available for training.

A standard CNN is constructed at first and additional layers are embedded to it, namely layers of deconvolution and unpooling. For the unpooling operation, the authors implement an approach in which the maximum pooling value is restored. The authors then perform unsupervised training of CAE by choosing a set of 5000 paintings from a resource that contains images with 1000x1000 pixel resolution.

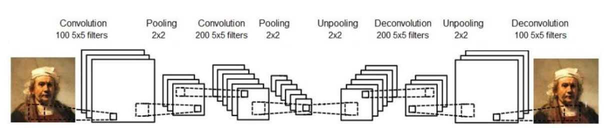

Figure. 1. CAE Architecture as in [1]

A standard CNN is constructed at first and additional layers are embedded to it, namely layers of deconvolution and unpooling. For the unpooling operation, the authors implement an approach in which the maximum pooling value is restored. The authors then perform unsupervised training of CAE by choosing a set of 5000 paintings from a resource that contains images with 1000x1000 pixel resolution. In Fig. 1., the various parameters along with their specific configuration of each layer.

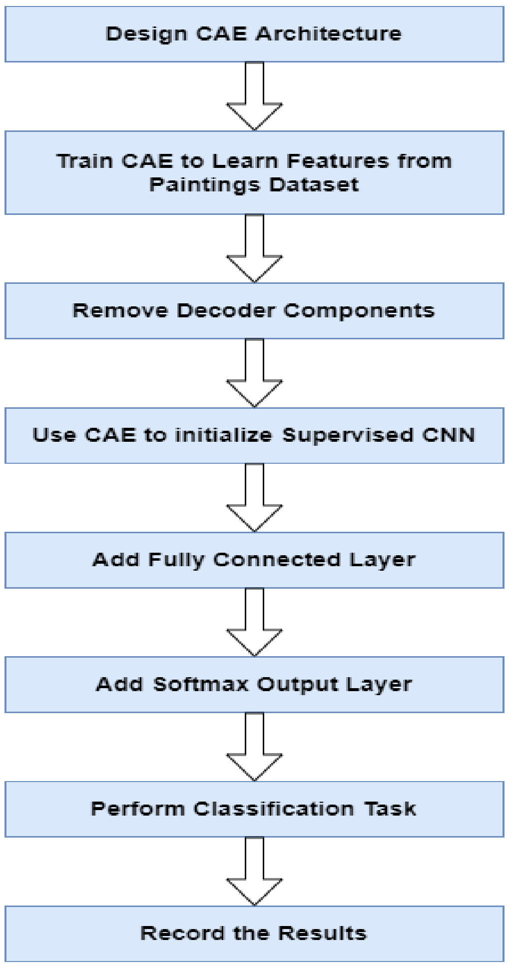

Figure. 2. Process flow in Paper[1]

The flowchart in Fig.2. gives an outline of the process flow that was adopted by the authors.

2.2 Key Results of the Original Paper

The results of the study in the original paper stipulate that the average accuracy obtained for the CNN on the validation set is significantly higher than those obtained by previous benchmark image processing and feature extraction techniques.

The following table shows the comparison of various techniques that the authors used along with their average accuracies. The accuracy of the Convolutional Autoencoder that they constructed is given in the last row.

Feature Extraction Method	Accuracy
Image Processing	70.79%
RBM	71.94%
Image Processing +RBM	76.93%
Convolutional Autoencoder	**96.52%**

Table 1: Summary of results in paper[1]

3. Methodology (of the Students' Project)

As discussed in the paper for a small number of training samples , autoencoders try to reconstruct the original input, so the initial input is mapped into a hidden layer and reconstructed to imitate the original input. They are trained by standard backpropagation to reduce this reconstruction error. The learned weights are used to in the following CNN model appended with a couple of dense layers and trained to give the final prediction

3.1. Objectives

The primary objective of the problem being, identifying the artist from the piece of artwork, the problem can be viewed as a problem of image authentication where the idea is to determine whether the artwork is by a particular artist or not, which is essentially a problem of binary image classification. The idea behind using an Auto Encoder as discussed in the paper is to shift the focus from primary image processing-based feature extraction, which are tailored to specific datasets to a more generalized approach which operates on the raw pixel level. We have attempted to implement the same model discussed in the paper and to take it a step forward by benchmarking the performance of this model with the VGG16, Xception, MobileNet and ResNet50 deep learning models.

3.2.Problem Formulation and Design Description

Although Convolutional Neural Networks outperform conventional image processing techniques, they require a larger dataset to train and learn all the parameters in the multi-stacked layers. They typically require images along the lines of a few thousands per class, but the fundamental limitation in the objective of the project is the number of artworks of the artist. So, the problem that we try to solve here is the effective utilization of the limited number of artworks by each artist. The authors of the paper constrain the number of artworks per artist that the model is trained upon and try to achieve the best results. The model is trained to classify paintings by Rembrandt, Renoir and Vincent Van Gogh. We conducted a study of the model described in the paper and attempted to expand the model by increasing the number of artists to 5 by including Pablo Picasoo and Francisco Goya. In addition to this, we conducted a similar study to the other deep learning models described above.

4. Implementation

The Convolutional Autoencoder consists of 2D convolutional layers followed by pooling layers. The mapped image is then passed through unpooling and deconvolution layers or transposed convolutional layers. Although the approach discussed in the paper implements max pooling and unpooling by storing the max locations, we implemented a much simpler standard UpSampling layer which applies the max value to all the locations in the grid. After the training of the Convolutional Autoencoder, we remove the upsampling and deconvolution layers and use the remaining part of the model to initialize the supervised CNN model with additional dense layers. The paper mentions 20% reduction in pixels to train the autoencoder to improve accuracy but they have not mentioned the layer in which this was implemented. We tried to implement the autoencoder with a dropout layer but since it degraded the performance, we decided to remove it. In addition to all the layers that were mentioned in the paper for the CNN, we had to add a flatten layer and then pass it to the softmax layer to classify the paintings.

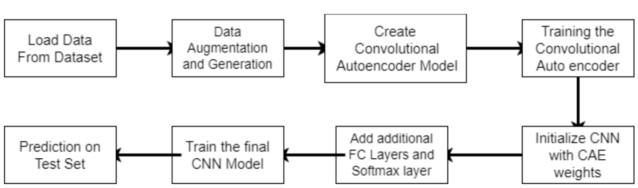

Figure. 3. Workflow in the model

4.1 Data

The Paintings dataset was obtained from Kaggle (https://www.kaggle.com/supratimhaldar/deepartist-identify-artist-from-art/data) since there were some issues in obtaining them from the Webmuseum link mentioned in the paper. The dataset consists of 50 Artists and each artist having around 100 to 300 paintings. The images are of varying resolution with approximately 1024 x 1024 pixels. For our study, we picked random paintings of Artists Rembrandt, Renoir and Van Gogh for the first part of the study and included paintings of Pablo Picaso and Francisco Goya for the second part of the study involving 5 artists. The images were rescaled to 256 x 256 pixels for our model.

4.2 Deep Learning Network

DeepPainter Model

The Autoencoder model used to train the data initially is shown below.

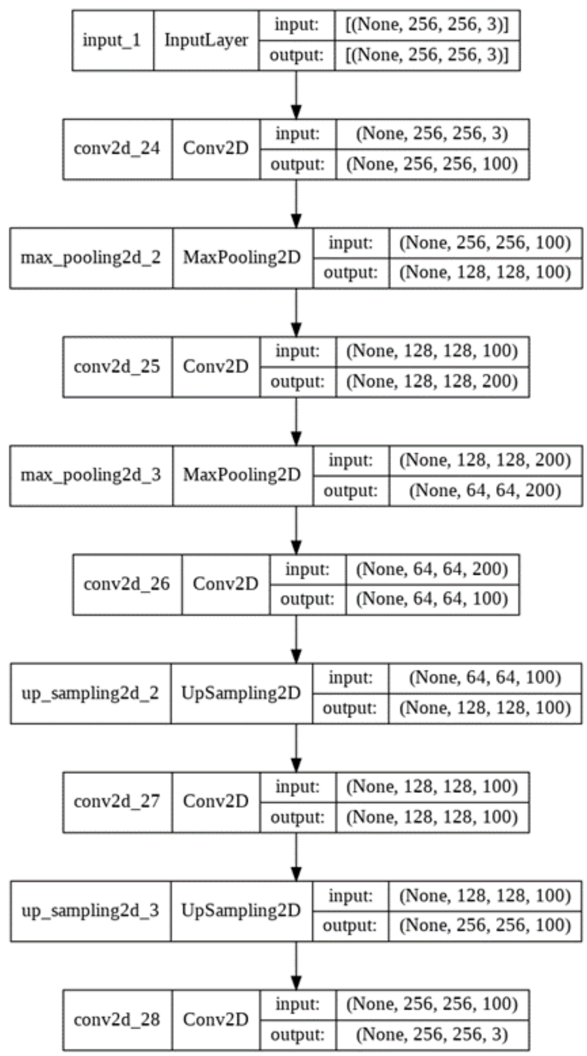

Figure.4. DeepPainter Autoencoder

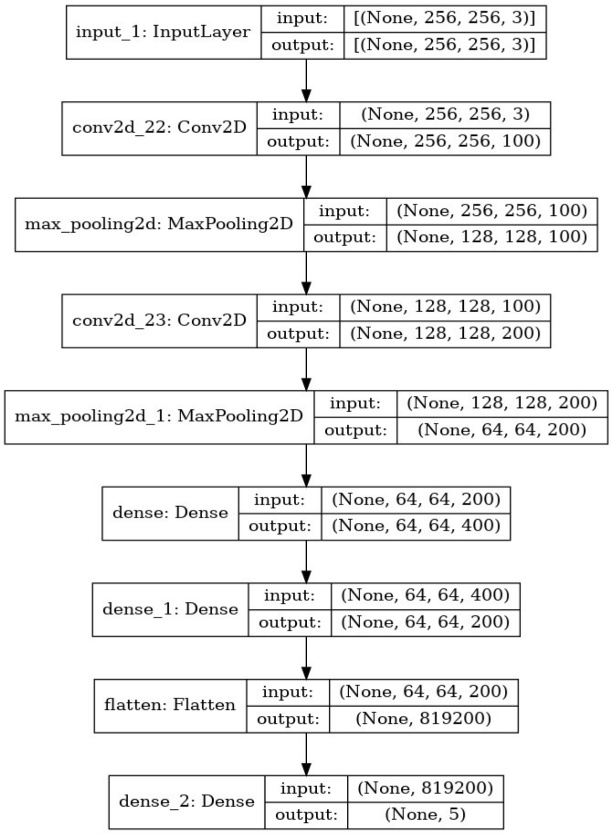

Figure.5. DeepPainter Convolutional Neural Network

4.2 Software Design

The Python codes for all models were implemented on Jupyter notebooks using TensorFlow version 2.4. These codes were run on Google Cloud VM Instance, running Ubuntu 18.04 LTS, using 2 Intel(R) Xeon(R) CPUs @ 2.00GHz and Tesla T4 GPU.

5. Results

5.1 Project Results

The findings of our study have been discussed in this section. The Convolutional Autoencoder model exhibited accuracies of 93%, 87% on the training, test set respectively. However, this result was achieved after increasing the dataset-size from 40 images per artist to 300. Prior to this, the accuracies fluctuated between 60% and 70% with no improvement upon varying the epochs, learning rate, optimizers.

The decoder was replaced with fully connected layers to carry out the artist-recognition. This model accuracy exhibited fluctuations between 70% and 80%, even with the increased data-set size. To ascertain the problem, several strategies were implemented – the Flatten () layer, which acts as a bridge between convolutional layers and dense layers, was placed at different locations. Similar to the Autoencoder, the values of the hyper-parameters were varied. The dataset was further increased to 500 per artist. None of the above modifications could improve the accuracy or reduce the noise.

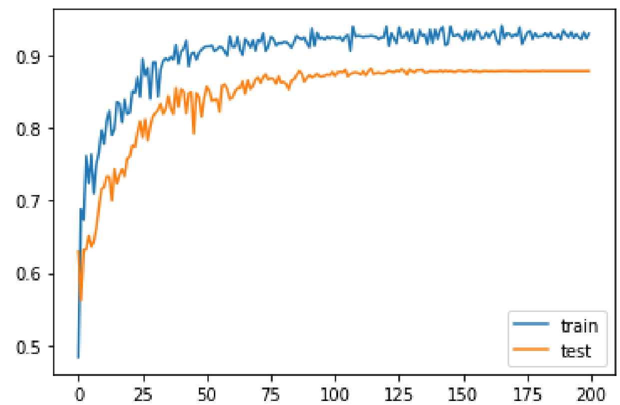

Figure 6. Accuracy Vs Epochs for the Convolutional Autoencoder (3-Artists)

The original dataset used by the author was derived from *webmuseum.meulie.net/wm* whereas the present work utilizes data obtained from *kaggle.com*. It can be concluded that the original model has been fine-tuned to perform well on a particular dataset and lacks generalization. This can be confirmed by analyzing the accuracies of the pre-existing complex CNN architectures such as MobileNet, VGG16 etc. which perform well on any datasets provided.

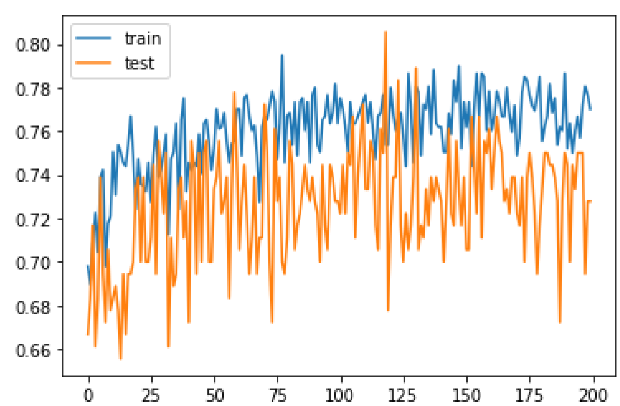

Figure 7. Accuracy Vs Epochs for the Artist-Classifier CNN (3-Artists)

Model	Training Accuracy	Test Accuracy	Training Time
Autoencoder	0.93	0.87	2m 39s
Artist-Classifier	0.78	0.73	23m 20s
VGGNet	0.87	0.82	23m 45s
Xception	0.86	0.85	29m 35s
MobileNet	0.9	0.83	22m 8s
ResNet50	0.68	0.65	23m 26s

Table 2. 3-Artist Classifier Accuracies

To test the robustness and generalisation behaviour of the current model, it was trained with a dataset of 5 artists' paintings, the results of which have been shown in Figures 8 ,9. Similar to the previous case of 3 artists, the Convolutional Autoencoder exhibits good performance here, with training and test accuracies of 0.9 and 0.87 respectively. On the other hand, the artist-classifier shows a decline in accuracy and no improvement in noise reduction. A similar trend was observed with the other CNN architectures whose accuracies reduced by an average of 16%. It can be inferred that the performance decline across all models is due to insufficient data i.e. they required more images per artist for more accurate training. Since the artist-classifier has not been optimized yet for a 3-artist classification, no conclusion can be drawn about its generalized behaviour. Figure 10. provides a comparative analysis of the 2 cases.

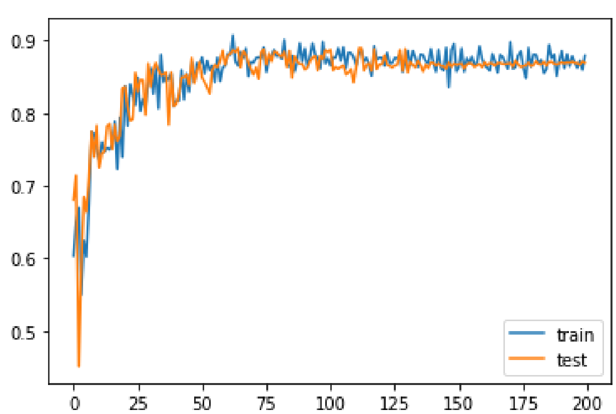

Figure 8. Accuracy Vs Epochs for the Convolutional Autoencoder (5-Artists)

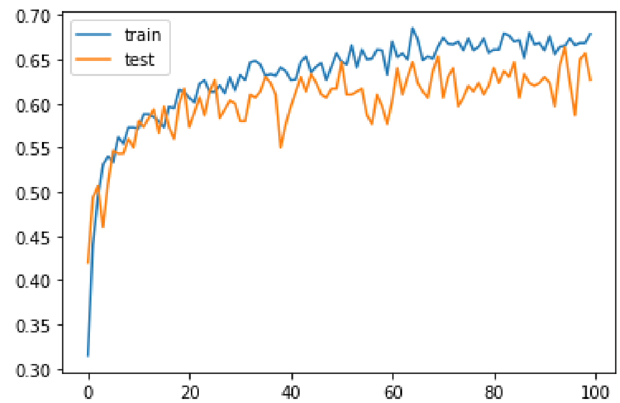

Figure 9. Accuracy Vs Epochs for the Artist-Classifier CNN (5-Artists)

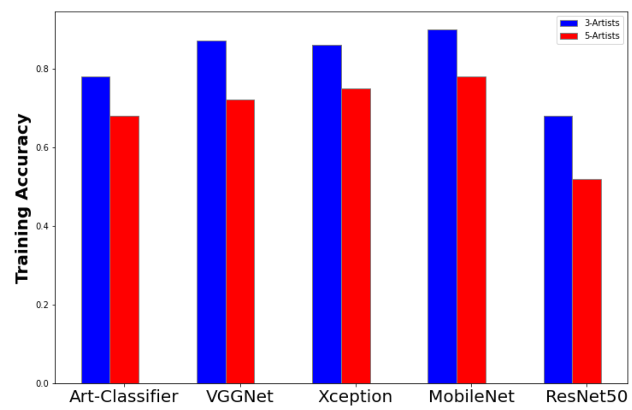

Figure 10. Comparison of Classifiers

Model	Training Accuracy	Test Accuracy	Training Time
Autoencoder	0.9	0.87	4m 19s
Artist-Classifier	0.68	0.63	38m 25s
VGGNet	0.72	0.67	39m 51s
Xception	0.75	0.66	39m 17s
MobileNet	0.78	0.73	37m 10s
ResNet50	0.52	0.53	38m 50s

Table 3. 5-Artist Classifier Accuracies

5.2 Comparison of the Results Between the Original Paper and Students' Project

The accuracy achieved by the author is 96.52%, as opposed to 73% in this work. Although the same model architecture was implemented, there are some key differences worth mentioning. As previously mentioned, a different dataset was used in the present study due to unavailability of the original dataset. Furthermore, we used 300 images-per-artist as opposed to 30. The author had specified a learning rate of 0.01 and reduction of 0.98 after every epoch. As opposed to this, the learning rate was initially set to 0.01 and gradually reduced to improve performance. Instead of every epoch, the learning rate was reduced by 0.9 every time a loss-plateau was encountered. The author split the data into 90% training and 10% validation data, while we implemented an 80-20 split.

The paper also mentions the use of a dropout of 20% of the pixels, use of a mask. However, there is a considerable amount of ambiguity in the description of the model, where the author has neglected essential aspects. We had to incorporate an additional Flatten() layer to transform the Conv2D() layer into a Dense() layer. No mention of such a layer or its location has been made. The difference in performances could be attributed to these differences in implementation. The paper claims to achieve accurate prediction using only 40 images per artist. Although this is commendable, there is a good possibility that the model has been designed to work well with that particular dataset. In other words, the model's generalized behaviour is questionable.

5.3 Discussion of Insights Gained

From the present study, it can be inferred that a general-purpose CNN classifier model, which works for any given dataset, would naturally be a very deep model with a large architecture. Smaller architectures may also perform exceedingly well, but only on a particular dataset. It was also noticed that as the number of classes increase, the dataset must be accordingly expanded for accurate classification. Finally, the tenacity of the model in providing the same accuracy despite varying learning rate etc. proves that hyper-parameter tuning comes secondary to obtaining the appropriate dataset and building a good deep-learning model.

6. Future Work

Although the accuracies obtained from the artist-classifier are satisfactory, further improvements can be implemented to achieve the same performance as mentioned in the author's original work. Furthermore, some aspects mentioned in the paper such as Dropout of 20% of the pixels, using a mask to store location of maximum etc. were not implemented in the current work. These could be incorporated in an attempt to improve performance. Another scope for further work is to incorporate additional regularization parameters such that the author's original model may perform well for any arbitrary dataset provided, similar to the pre-existing CNN models used in this study.

7. Conclusion

This project deals with the implementation of a Convolutional Autoencoder, whose encoder is used to initialize a supervised CNN in order to perform artist-classification based on their paintings. The model was constructed as per the specifications provided by the author, with some minor tweaks with regard to layer flattening, learning rate as well as use of a different dataset. With an initial dataset of 40 images-per-artist, there was a poor performance of both Autoencoder and Classifier, out of which the former's performance was improved upon increasing the dataset size. The classifier exhibited considerable noise in accuracy values, which oscillated between 70 and 80%, which did not improve with data or hyper parameter tuning.

It was therefore concluded that the original model was built for a specific dataset and lacked generalized behaviour. This was confirmed by benchmarking the performance against well-known CNN classifiers such as MobileNet, VGGNet, Xception and ResNet50, which exhibited accuracies of up to 90%. In order to further test this generalized behaviour, the model was made to classify paintings from 5 artists, but suffered a decline in performance. While such a decline was observed across all models, the limitations of the author's model cannot be ruled out. Regarding the distinct differences in accuracies between the author's work and the present work, 96.25% and 73% respectively, such a disparity could be attributed to the ambiguity, lack of conciseness in the author's description of the model.

6. Acknowledgement

The successful completion of this work would not be possible without the abundant online resources available on medium.com, stackexchange.com etc. We would like to take this opportunity to express our gratitude to the teaching assistants who were prompt with clarifications and any other miscellaneous assistance required. Most of all, it was the extensive knowledge and insight obtained from Prof. Kostic's lectures that enabled us to effectively build and train our own models.

7. References

[1] Eli (Imid) David, Nathan S. Netanyahu, "Deep Painter: Painter Classification Using Deep Convolutional Autoencoders", International Conference on Artificial Neural Networks (ICANN), Springer LNCS, Vol. 9887, pp. 20–28, Barcelona, Spain, September 2016.
[2] Jonathan Masci, Ueli Meier, Dan Ciresan, and Jurgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In International Conference on Artificial Neural Networks, ICANN, pages 52–59, 2011.
[3] M.D. Zeiler, G.W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In International Conference on Computer Vision, ICCV, pages 2018–2025, 2011.
[4]M.D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In European Conference on Computer Vision, ECCV, pages 818–833, 2014.

8. Appendix

The code for this project can be found at https://github.com/ecbme4040/e4040-2021fall-project-MSKD-km3714-mr4136-sc4921

8.1 Individual Student Contributions in Fractions

	UNI1	UNI2	UNI3
Last Name	Sreenivasa	Rajkumar	Chintala
Fraction of (useful) total contribution	1/3	1/3	1/3
What I did 1	Constructed the model as per the specification of the author's paper. Improved the accuracy of the Autoencoder, ran all the models for 5 artist-classification, gathered all plots and figures and authored the results section		
What I did 2	Constructed the model as per the specification of the author's paper. Coded the dataset generation, pre-processing, data augmentation. Wrote the code for training the autoencoder and plots. Authored the sections – Objectives and Challenges, Problem Formulation, Implementation		
What I did 3	Used transfer learning methodology (like VGG16, MobileNet, Xception and ResNet50) to compare the models and increased the accuracy. Constructed and ran all the models for 3 artist-classification. Authored the Abstract, Introduction, Description of the Original Paper and Present Work.		