RetinoPath

Luisa Ortner
luisa.ortner@tum.de

Francesca D'Amico
francesca.damico@tum.de

Yiwen Liu
yiwen.liu@tum.de

Chair of Computer Aided Medical Procedures (Prof. Nassir Navab)
Practical Course: Machine Learning in Medical Imaging 2023 SS

Abstract

RetinoPath aims to use state-of-the-art deep-learning-based methods for retinal disease progression analysis using longitudinal sequential images as input. To unveil the complex patterns and mechanisms underlying various retinal disorders, we first analyze and summarize the related work and datasets in this field and search for several baselines to be the suitable candidate for this task. Since we cannot access any dataset in this field, we generate synthetic datasets to conduct some ablation studies of our baselines. The main objective is to carry on a series of experiments and make comparisons of different model architectures using possible baselines to seek the opportunity to improve early detection and predict disease progression trajectory.

1 Introduction

Disease progression trajectory prediction aims to forecast future clinical conditions of patients based on longitudinal data, such as electronic health records (EHRs), biomarkers and medical images.[1] The objective is to provide medical personnel with predictive tools that can support disease diagnosis, enhance treatment strategies, assist clinical decisions, and improve clinical outcomes.[2]

A broad-spectrum of methodologies have been applied to the disease prediction field, from logistic regression to more sophisticated deep learning models, also techniques such as ensembles models and trajectory clustering. Most of these models employ comprehensive longitudinal EHR data or medical codes.[3] Data is a key aspect for machine learning models, as longitudinal medical data has several challenges. In fact a comprehensive model should be able to deal with problems such as missing data, data heterogeneity, unregular time intervals between sessions, and high dimensionality.

The goal of this project is to adapt existing image prediction methods to longitudinal retinal images, resulting in future images that doctors can use to make more informed decisions. In addition to this, complementary techniques, such as disease classification and severity, were explored to lay the foundation for a more comprehensive future methodology.

2 Related Works

As we discuss above, there are various different inputs that can be used for retinal disease progression prediction. For this task and similar fields, most works use medical codes (including EHR data and bio-markers)[4] or a single image as input to make some prediction[5]. Due to space reasons, in this part, we mainly focus on works using sequential images as input. We will answer three main issues: how to encode the input sequential images, which kind of sequential models to use, and which kind of prediction we want to know.

2.1 Encode the Input Images

About how to transfer images to the feature embeddings that the model can accept, we summarize the methods shown in Figure 1. There are two general methods: "whole image processing" and "patch partition". While the first one means the unit where we process images is at least using a single whole image, which is in contrast to the patch division. The patch partition will divide an image into multiple small pieces.[6]

The "whole image processing" methods are very straightforward and easy to implement. As for each single image, we can use a classical backbone like VGG16 [7] or ResNet101 [8] to separately extract the sequential feature vectors or feature maps. The temporal information will be captured by the following sequential models. In contrast, the 3D Convolution [9] gives us the chance to encode the image inputs considering the temporal information in the mean time. However, we need to consider how to design

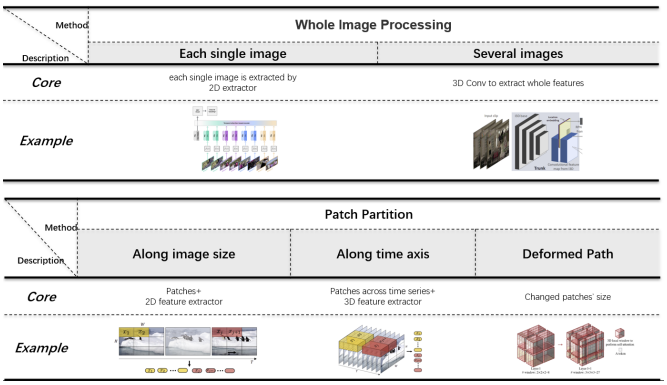

Figure 1. Summary of methods to encode the input images.

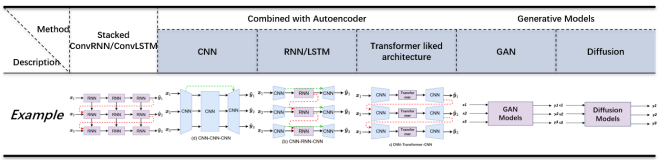

Figure 2. Summary of different sequential models.

the sequential model architectures to process single feature map output.[10]

The "parch partition" arises from the vision transformer, we divide a single image input into several small blocks, and then extract features of each blocks as the tokens.[6] The first method of this is a very direct way to divide each sequential image into several patches and concatenate them together[11], but it will be very computationally consuming. One step further to improve the computation efficiency[11] is to divide the patches along with the temporal axis. However this method still lacks the information exchange between the neighboring patches, so we introduce a changed patch size to get the connection between different patches without more computation cost.[12]

For our baseline implementation, we finally choose the Autoencoder to get the bottleneck layer as the following model's input.

2.2 Sequential Models

After getting the embeddings of the inputs, we need a sequential model to process these embeddings. The sequential model here is a broad concept, which means the model can process sequential input. The classical model concept like RNN [13], LSTM [13] and Transformer [14] are included. We generally categorize the models into three parts: Stacked RNN/LSTM, models combined with Autoencoder, and generative models as shown in Figure 2.

As the name means, Stacked RNN/LSTM uses several RNN/LSTM blocks to process the sequential input, and to adapt the image input, we should use the convolution network as an internal component of RNN or LSTM [15]. Similar to this kind of architecture, we can directly use RNN or LSTM models with an Autoencoder model to process sequential information [16]. Due to the training of RNN and LSTM not being in parallel, and the receptive field and process sequence length are limited, the more popular architecture transformer [14] can also be used for sequential input. The only problem with the transformer is training process is very time and resources expensive[17]. As a result, some small and robust CNN models[18] also perform very well. To using CNN to process the temporal information, we need to specially design CNN layers. You can check our baselines and their categories in Section 4.

2.3 Process the Output

One important variable in the disease progression is which kind of indicators are used to check the future possible development. In the disease prediction field, the output of the model can be categorized into disease sequence prediction, disease trajectory forecasting [19], severity or comorbidity risk prediction [20], and future disease images. There are several methods to process the output features of a model, simply we can add some linear layers [21] or attention layers [22] to fusion the information and get the specific variable we want to get, such as the severity or comorbidity risk and the parameters of disease trajectory curve[20].

Since there are not many models for predicting disease trajectories, we decided to focus on predicting future disease images. Besides, the output of future images can also help us interpret the model. The goal would be to classify these images according to disease severity at a later stage.

3 Datasets

When it comes to longitudinal retinal datasets, there are a few problems. The first is that there are no public datasets. While there are some public retinal datasets, they are not longitudinal. Moreover, there are very few datasets with additional patient information. Some authors are not interested in sharing their dataset. And lastly, there are privacy issues and legal restrictions with sharing medical datasets. Following is a list of interesting (non-public) longitudinal datasets.

- HARBOR study (monthly spectral-domain OCT for AMD, 1097 patients, 24 images each)[23]

- RISE and RIDE datasets (stereoscopic 7-field CFPs, 764 patients, 3 each)[24]
- INSIGHT datasets (13 datasets, OCT and fundus images with additional patient information[25]
- Moorfields exAMD dataset (8692 patients, 1994798 images, fundus and OCT images, AMD)
- Klinkum rechts der Isar dataset (OCT and fundus images with patient information)

Since the process of requesting access to datasets takes long, the ChestX-ray8[26] dataset was used for intermediate testing. It is a Chest X-ray Dataset of 14 Common Thorax Disease Categories and includes 112.120 images with additional patient information like age and gender. Since the purpose of this project was to predict the disease progression with longitudinal images and not to classify the disease, the data was cleaned to restrict the dataset to only one disease pattern and every patient has five images, where four images are used as input and the fifth is used for prediction. The final dataset contains 4245 images of 849 patients.

Since we were not provided with a longitudinal retinal dataset, we created two synthetic datasets. For each dataset 30 images were generated and 20 of them randomly selected for the dataset. This should simulate unregular time intervals in real medical datasets. The eye and disease position is random for each patient and each patient has a different disease growing rate. Each dataset contains 10000 patients. For each patient 10 images were used as input and 10 were predicted.

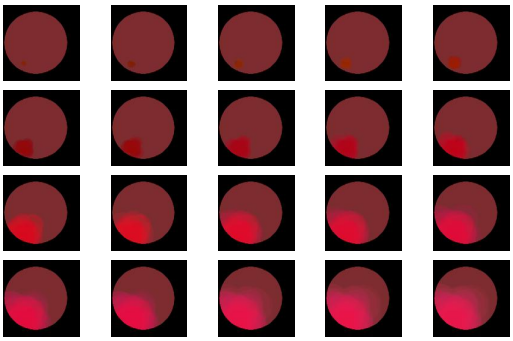

Figure 3. One sample of synthetic dataset 1.

The first synthetic dataset should model different progression of the AMD disease for different patients. The form of the disease is random growing and the opacy which is added for each time step decreases with time. The severity and the spread of the disease varies between patients. One sample can be seen in Figure 3.

The purpose of the second dataset was to create images with growing circles with changing colors to evaluate whether models can understand this simple development. One sample can be seen in Figure 4.

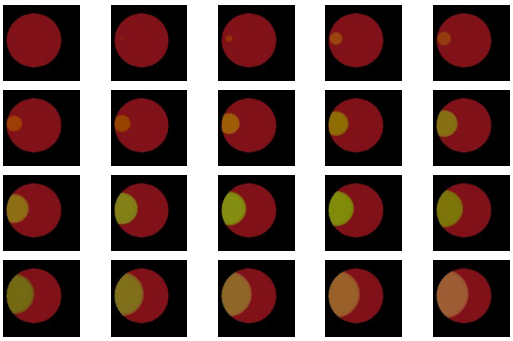

Figure 4. One sample of synthetic dataset 2.

Table 1. Baselines.

Class	Method	Input/Prediction
AE + CNN	SimVP[18]	Images/Images
AE + CNN	WALDO[27]	Images/Images
AE + LSTM	LMC[28]	Images/Images
AE+Trans.	VPTR[17]	Images/Images
AE+Trans.	ClinicalGAN[19]	Medical code
Foundation	RetFound[29]	Images/Disease class
GAN	StyleGAN2[30]	Images/Images
Diffusion	SADM[31]	Images/Images
Diffusion	RVD[32]	Images/Images

4 Baselines

In Table 1 an overview of all baselines can be seen. Taking into account model category comprehensiveness, we also consider choosing more models that may have more advantages. Only ClinicalGAN, RetFound, and SADM are medical models, the others are models for predicting future video frames and have been adapted for the medical case. The details of each baseline will be introduced briefly in Section 5, and the illustration images of each baseline can be found in Appendix A. We conduct the Experiments on these baselines using ChestX-ray8[26], Synthetic dataset 1, and Synthetic dataset 2. For the ChestX-ray8[26] dataset, we input 4 past images to output 1 future image. And for Synthetic datasets 1 and 2, we input 10 past images to output 10 future images.

5 Experiments

5.1 Evaluation Metrics

In the context of predicting future medical images from longitudinal data, the evaluation of models is crucial for

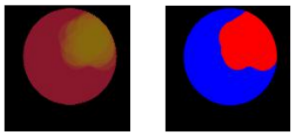

Figure 5. One sample of the segmentation map.

Figure 6. One sample of the flow map.

their clinical utility. Mean Squared Error (MSE) measures the average squared difference between the predicted and actual pixel values, highlighting overall accuracy but not necessarily perceptual quality. Structural Similarity Index (SSIM) assesses the perceptual similarity in terms of luminance, contrast, and structure, which are essential for evaluating the perceptual quality and clinical relevance of medical images. Peak Signal-to-Noise Ratio (PSNR) quantifies image fidelity and is especially relevant when preserving fine details is critical, such as in medical images. Learned Perceptual Image Patch Similarity (LPIPS) measures the distance between feature representations at different scales, which is crucial in medical imaging since important details are at varying levels.

5.2 Sub-optimal Baseline Results

5.2.1 WALDO

For WALDO segmentation and flow maps are needed as input. The segmentation maps are created during the creation of the datasets. The disease area is segmented in red, the eye in blue and the background in black as can be seen in Figure 5. For the creation of the flow maps the function calcOpticalFlowFarneback of cv2 was used and the result can be seen in Figure 6.

5.2.2 SADM

SADM is a sequence-aware diffusion model, which is optimized for the medical case, as it is robust towards sequences with various lengths, missing data or frames and high dimensionality.[31] It was not capable of learning the dataset representation. The reason could be changes in the model, as the original model accepted only 3D inputs but the images used in this project were 2D. Another reason could be that the published code was only the minimal code

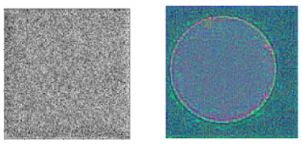

Figure 7. Prediction of SADM on the left and prediction of RVD on the right.

and the author stated that the published model would therefore not give the same result as in the publication[31]. Predictions of SADM only consists of noise as can be seen in Figure 7 on the left.

5.2.3 RVD

RVD[32] combines RNNs with a diffusion probabilistic model. It successively generates future images by first using RNNs to predict the next frame and then corrects this frame with a stochastic residual generated by an inverse diffusion process. RVD is only capable of predicting the eye shape and cannot identify the disease area, as can be seen in Figure 7 on the right.

5.2.4 RETFound

RETFound [29] is a foundation model for retinal images that learns generalizable representations from unlabeled retinal images and is adaptable for various applications, including ocular disease classification and severity prediction. The methodology comprises in two stages. In the first stage, RETFound undergoes pretraining via SSL, employing the masked autoencoder. During this phase, the model acquires representations through a pretext task, enabling it to capture retina-specific context, including vital anatomical structures that serve as potential indicators for neurodegenerative and cardiovascular diseases. In the second stage, the acquired knowledge is utilized to fine-tune the pretrained model for specific downstream disease detection tasks.

We conducted RETFound fine-tuning using our synthetic dataset with the objective of predict disease severity. However, it is important to note that due to time constraints, we were unable to complete an exhaustive training process. Upon evaluation, the model achieved a categorical accuracy of 0.5756 on the test dataset. It is essential to consider the limitations imposed by the synthetic dataset and shortened training duration. Further refinements in model parameters and additional training could potentially lead to improved predictive performance.

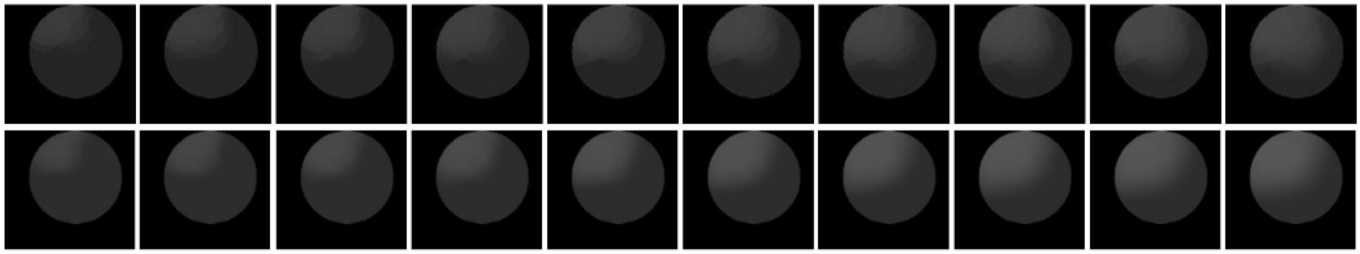

Figure 8. Prediction of one sample (above) compared the ground truth (below).

Table 2. Evaluation metrics on test set after different number of epochs of synthetic dataset 1.

# of epochs	MSE↓	PSNR↑	LPSIS↓	SSIM↑
50	62.766	24.849	0.327	0.785
100	46.225	27.514	0.203	0.745
1000	10.907	34.783	0.020	0.970
10000	8.689	36.460	0.019	0.974
100000	**8.227**	**37.038**	**0.012**	**0.978**

Table 3. Evaluation metrics on test set after different number of epochs of synthetic dataset 2.

# of epochs	MSE↓	PSNR↑	LPSIS↓	SSIM↑
50	56.075	25.559	0.328	0.782
100	55.113	26.495	0.269	0.741
1000	22.801	31.991	0.038	0.940
10000	**15.353**	**34.225**	0.029	0.963
100000	18.045	34.038	**0.026**	**0.964**

5.3 Optimal Baseline Result with Ablation Studies

5.3.1 LMC

LMC[28] is a video prediction model which recalls long-term motion context via memory alignment learning. During training it stores the long-term motion contexts into the memory and during testing, the input sequences are matched with the sequences in the memory. Additionally it incorporates and ConvLSTM to predict the future images. It was trained and tested on both dataset 1 and dataset 2. After different number of epochs the model was tested on the hold-out test set and the quantitative results can be seen in Table 2 and Table 3. The best trade-off between training time and accuracy is 1000 epochs for both datasets. The qualitative results for one sample can be seen in Figure 8 for dataset 1. As can be observed the results are accurate as the model is capable of identifying the correct disease area and severity.

5.3.2 SimVP

SimVP[18] means "Simpler yet Better Video Prediction", the authors want to show that some smaller models can also outperform the large and complex models. As a result, this model uses pure CNN architectures to process the sequential image input. From the Figure 13 from Appendix A,

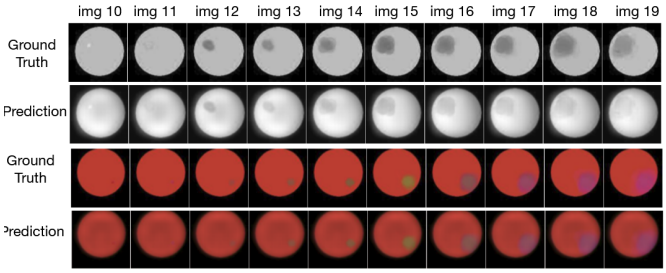

Figure 9. The best output experiment result of SimVP model.

the model deploys the Autoencoder architecture to extract feature embeddings. The core here is to design a pyramid connection form using Inception modules to only convolute T × C channels on (H, W). In this way, it can learn temporal evolution.

We conduct some ablation studies within the SimVP[18] model using Synthetic Dataset 1. Following the controlled variable principle, we conduct a total of four groups of ablation studies to show the effect of different input data augmentation, training samples, and learning rates in Table 4. For the SimVP[18] model, colored or gray inputs can both achieve decent and comparable quantitative results. Note that the MSE criterion will calculate the sum number of

three channels for the colored input, so the numbers are larger than gray images. Besides, it is not necessary for this kind of smaller CNN model to input large amounts of samples. What's more, we observe that the results don't get better after around 200 epochs, so we consider the training loss back-propagation oscillating, and we add a learning rate scheduler to let the loss decrease more. The prediction sample is shown in Figure 9, the disease area shape and growth tendency are almost right.But the quantitative results are not good as LMC in Section 5.3.1.

5.3.3 VPTR and ClinicalGAN

The baselines VPTR[17] and ClinicalGAN[19] are both based on transformer[14] architectures combined with Autoencoder as explained in Section 2. The training process is divided into two stages and therefore we will introduce these two baselines together. The first training stage is ResNet-based Autoencoder training shown in Figure 10. Note the difference here for these two models is the size of bottleneck layer. The VPTR transformer[17] has special spatial-temporal attention layers, so the bottleneck is a feature map. However, the ClinicalGAN[19] model uses a classical transformer and the input token should be a feature vector, so we need an additional linear layer or average pooling layer to map the feature maps to vectors.

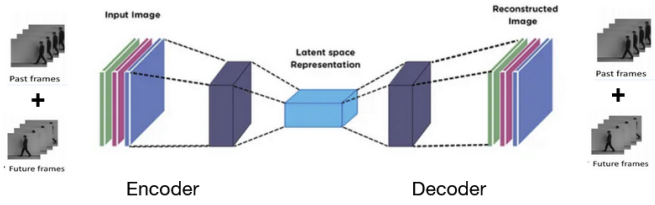

Figure 10. The first training stage of VPTR and ClinicalGAN.

The results of ablation studies can be found in the Table 5 and Table 6 in the Appendix C. The organization is similar to the SimVP[18] baseline. But for the VPTR[17] and ClincialGAN[19], the grayed images can achieve much better results than the colored images. We analyze the reasons due to the model will more focus on learning the colors instead of the area and trend of the disease. For the Autoencoder training, training with or without GAN framework do not have any effect. Also for the ClinialGAN Autoencoder, we mainly investigate how to map the feature maps to feature vectors. The quantitative result shows the linear layer can keep more information and is slightly better than using the average pooling. Besides, due to the information loss during the mapping process, we must use a large amount of data to let the model learn more knowledge about the dataset. We choose the best Autoencoder model and freeze the weights to train the following transformer models. The intuitive results can be found in Figure 21 and Figure 23 from Appendix B.

The second training stage is transformer training. As we talk above, the authors of VPTR design special attention layers to achieve efficient transformer calculation for spatial—temporal input. The first attention layer operates on the spatial size like the vision transformer[6], it divides the feature map into several patches and calculates attention. The second attention layer is along the temporal sequence to capture the relationship between the sequential inputs. For the ClinicalGAN baseline, the novelty is the model combines the transformer and GAN architecture like TransGAN[33] architecture. The generator of GAN is the encoder and decoder of the transformer, the discriminator only uses the encoder part of the transformer to identify the future ground truth images and predicted future images, which is shown in Figure 11.

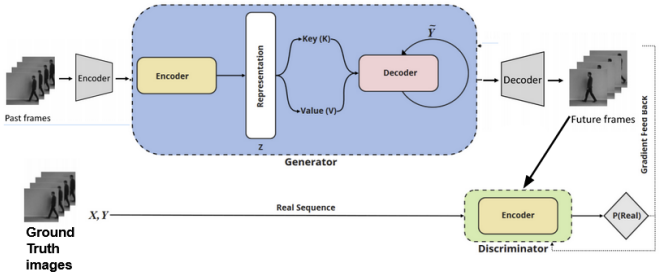

Figure 11. The second training stage of VPTR and ClinicalGAN.

As the ClinicalGAN originally uses medical codes as input and output, we modify the data loader, model architecture, loss functions, and training script to adapt to the image input. We add the GAN training framework to the VPTR to let the model output better-quality images instead of repeated future images.[17] We do not have many good results for this training part. You can check intuitive results from Figure 22 from Appendix B. As before, we conducted several experiments to investigate how to output good results. In the experiments, we find transformer training is much more data-hungry than our imagination. So without a large amount of input data, we are not able to get good results, which can be checked by Table 7 and Table 8 in the Appendix C.

In the tables, we can see the quantitative result increases a lot with more training data. The result is shown in Figure 12. However, with a large amount of data, it is difficult to train the transformer from scratch. One problem is

Table 4. Ablation studies within SimVP[18] model using synthetic dataset 1.

Item	Input	Patients Number	Epochs/Training Time	Batch Size/Learning Rate	MSE↓	PSNR↑	LPSIS↓	SSIM↑
1	Colored images	1000	2000 epochs/13 h on a RTX 3090 GPU	32/Fixed = 0.02	620.5929	19.8893	0.1184	0.7581
2	Gray images	10000	2000 epochs/13 h on a RTX 3090 GPU	32/Fixed = 0.02	290.3340	18.3301	0.1770	0.7068
3	Gray images	1000	2000 epochs/70 h on a RTX 3090 GPU	32/Fixed = 0.02	285.6044	18.5301	0.1745	0.7203
4	Gray images	1000	2000 epochs/13 h on a RTX 3090 GPU	32/Scheduler every 200 epochs from 0.02	**271.1421**	**18.3353**	0.2599	0.5978
5	Gray images	1000	2000 epochs/13 h on a RTX 3090 GPU	32/Scheduler every 200 epochs from 0.02	610.1958	19.7455	**0.1708**	**0.6960**

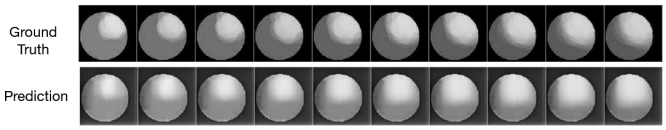

Figure 12. The best result of VPTR whole model.

the training is very time and resource-consuming: the final training of the VPTR transformer costs 150 h on 2 RTX 2080Ti GPUs and we can only choose batch size equal to 1 on one GPU. We cannot achieve large batch sizes as authors without training resources. Another problem is the VPTR model has 100 million parameters and the ClinicalGAN model has 1 million parameters. With more parameters, it is very hard and tricky to tune the hyper-parameters for these two models. As a result, we still use the hyperparameters from the original baseline, although these hyperparameters are very close to the original dataset. In addition, the GAN loss and contrastive loss also make the training very unstable. With inappropriate scale of these two losses will instead harm the training process. In the ablation studies, we find using original coefficients can lead to a very small improvement in results. One thing that can be confirmed is using a designed transformer layer can outperform the classical transformer layers.

5.4 Future Feasible Method: StyleGAN2

Absence of realism within the data significantly constrains the practicality and utility of the model. Thus, we have experimented with generative methods to create synthetic data endowed with authentic visual features. We planned to employ non-longitudinal retinal images to create a latent space, and subsequently use a generator to synthesize the next image given the input one.

Predicting Osteoarthritis (OA) Progression in Radiographs via Unsupervised Representation Learning [30] is an unsupervised learning approach to predict the future development of Osteoarthritis based on radiographs. Their approach consists in generate future synthetic images and then infer OA progression risk on them. They employ StlyeGAN2 [34] to learn representations of radiographs. Then, in order to find the vector that best matches the image they invert the StyleGAN generator.[30] Finally, by incorporating radiograph's time tamps, a latent vector field, $G^{-1}(basline) \rightarrow G^{-1}(followup)$, is constructed.

Our StyleGAN representation effectively captures structural and shape variations regarding the disease in our synthetic dataset. However, a notable issue is inconsistent color representation within the disease area. While the model captures the spatial characteristics of the anomalies, it occasionally generates diseases with mixed colors. This inconsistency in color representation is an aspect that needs to be addressed.

6 Outlook

Our research was conducted using synthetic datasets, which, while informative, may not fully capture the complexities of real retinal disease progression. In the future, obtaining access to a retinal longitudinal dataset or creating a more realistic synthetic dataset through generative models could enhance the relevance and practicality of our research.

The medical community is still unclear about how the AMD disease progresses. Especially since it converges to either wet or dry AMD at a later stage. It is particularly interesting for physicians to identify at an early stage to

which class it will converge. Machine learning models capable of generalizing retinal disease progression have the potential to make significant contributions to this area of research. Physicians and researchers could in fact leverage these models to gain deeper insights into the temporal evolution of retinal diseases.

Furthermore, integrating demographic data, medical history, genetic markers, and other patient-specific information into our models can provide a more comprehensive view of disease progression, enabling personalized predictions and treatment recommendations. Especially since it is possible for different disease trajectories to be grouped according to specific patient information.

As previously mentioned, some challenges commonly found in longitudinal medical datasets are high dimensionality, and data heterogeneity. Addressing these challenges will expand the applicability of our models in real-world medical scenarios.

References

[1] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, M. Hardt, P. J. Liu, X. Liu, J. Marcus, M. Sun *et al.*, "Scalable and accurate deep learning with electronic health records," *NPJ digital medicine*, vol. 1, no. 1, p. 18, 2018.

[2] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, "Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis," *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604, 2017.

[3] S. Mullin, J. Zola, R. Lee, J. Hu, B. MacKenzie, A. Brickman, G. Anaya, S. Sinha, A. Li, and P. L. Elkin, "Longitudinal k-means approaches to clustering and analyzing ehr opioid use trajectories for clinical subtypes," *Journal of Biomedical Informatics*, vol. 122, p. 103889, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1532046421002185

[4] C. A. Stevens, A. R. Lyons, K. I. Dharmayat, A. Mahani, K. K. Ray, A. J. Vallejo-Vaz, and M. T. Sharabiani, "Ensemble machine learning methods in screening electronic health records: A scoping review," *Digital Health*, vol. 9, p. 20552076231173225, 2023.

[5] A. Konwer, X. Xu, J. Bae, C. Chen, and P. Prasanna, "Temporal context matters: Enhancing single image prediction with disease progression representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 18 824–18 835.

[6] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," *ACM computing surveys (CSUR)*, vol. 54, no. 10s, pp. 1–41, 2022.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[10] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 244–253.

[11] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6836–6846.

[12] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3202–3211.

[13] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," *Advances in neural information processing systems*, vol. 28, 2015.

[16] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3163–3172.

[17] X. Ye and G.-A. Bilodeau, "Vptr: Efficient transformers for video prediction," in *2022 26th International Conference on Pattern Recognition (ICPR)*, 2022, pp. 3492–3499.

[18] G. Zhangyang, C. Tan, L. Wu, and S. Li, "Simvp: Simpler yet better video prediction," pp. 3160–3170, 06 2022.

[19] V. Shankar, E. Yousefi, A. Manashty, D. Blair, and D. Teegapuram, "Clinical-gan: Trajectory forecasting of clinical events using transformer and generative adversarial networks," *Artificial Intelligence in Medicine*, vol. 138, p. 102507, 02 2023.

[20] B. Lim and M. van der Schaar, "Disease-atlas: Navigating disease trajectories using deep learning," in *Machine Learning for Healthcare Conference*. PMLR, 2018, pp. 137–160.

[21] Y. Yuan and L. Lin, "Self-supervised pretraining of transformers for satellite image time series classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 474–487, 2020.

[22] S. Ging, M. Zolfaghari, H. Pirsiavash, and T. Brox, "Coot: Cooperative hierarchical transformer for video-text representation learning," *Advances in neural information processing systems*, vol. 33, pp. 22 605–22 618, 2020.

[23] A. C. Ho, B. G. Busbee, C. D. Regillo, M. R. Wieland, S. A. Van Everen, Z. Li, R. G. Rubio, and P. Lai, "Twenty-four-month efficacy and safety of 0.5 mg or 2.0 mg ranibizumab in patients with subfoveal neovascular age-related macular degeneration," *Ophthalmology*, vol. 121, no. 11, pp. 2181–2192, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0161642014004291

[24] P. G. S. A. M. K. M. H. L. H. L. T. Michael Singer, Mimi Liu and Z. Haskova, "Predictors of early diabetic retinopathy regression with ranibizumab in the ride and rise clinical trials," *Clinical Ophthalmology*, vol. 14, pp. 1629–1639, 2020. [Online]. Available: https://www.tandfonline.com/doi/abs/10.2147/OPTH.S247061

[25] I. M. E. H. N. F. Trust. (2023) The health data research hub for eye health. [Online]. Available: https://www.insight.hdrhub.org/

[26] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017. [Online]. Available: https://doi.org/10.1109%2Fcvpr.2017.369

[27] G. L. Moing, J. Ponce, and C. Schmid, "Waldo: Future video synthesis using object layer decomposition and parametric flow prediction," 2023.

[28] S. Lee, H. G. Kim, D. H. Choi, H.-I. Kim, and Y. M. Ro, "Video prediction recalling long-term motion context via memory alignment learning," 2021.

[29] Y. Zhou, M. A. Chia, S. K. Wagner, M. S. Ayhan, D. J. Williamson, R. R. Struyven, T. Liu, M. Xu, M. G. Lozano, P. Woodward-Court *et al.*, "A foundation model for generalizable disease detection from retinal images," *Nature*, pp. 1–8, 2023.

[30] T. Han, J. N. Kather, F. Pedersoli, M. Zimmermann, S. Keil, M. Schulze-Hagen, M. Terwoelbeck, P. Isfort, C. Haarburger, F. Kiessling *et al.*, "Predicting osteoarthritis progression in radiographs via unsupervised representation learning." *CoRR*, 2021.

[31] J. S. Yoon, C. Zhang, H.-I. Suk, J. Guo, and X. Li, "Sadm: Sequence-aware diffusion model for longitudinal medical image generation," in *Information Processing in Medical Imaging*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:254823541

[32] R. Yang, P. Srivastava, and S. Mandt, "Diffusion probabilistic modeling for video generation," 2022.

[33] Y. Jiang, S. Chang, and Z. Wang, "Transgan: Two pure transformers can make one strong gan, and that can scale up," *Advances in Neural Information Processing Systems*, vol. 34, pp. 14 745–14 758, 2021.

[34] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *Advances in neural information processing systems*, vol. 33, pp. 12 104–12 114, 2020.

A The baseline architectures

A.1 SimVP

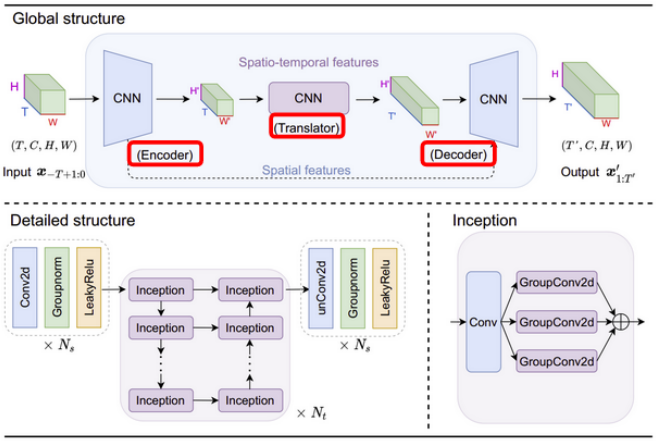

Figure 13. SimVP[18] model architecture.

A.2 WALDO

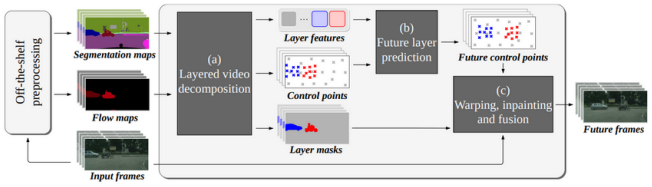

Figure 14. WALDO[27] model architecture.

A.3 LMC

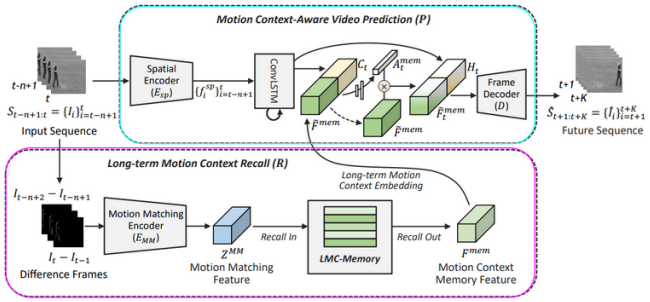

Figure 15. LMC[28] model architecture.Red: long-term motion context recall from external memory (LMC-Memory). Pink : LSTM predicts future frames considering recalled long-term motion context

A.4 VPTR

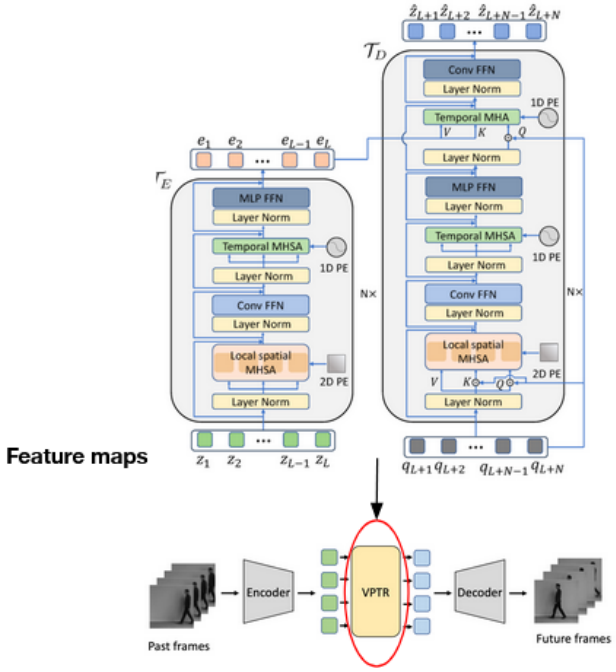

Figure 16. VPTR[17] model architecture.

A.5 ClinicalGAN

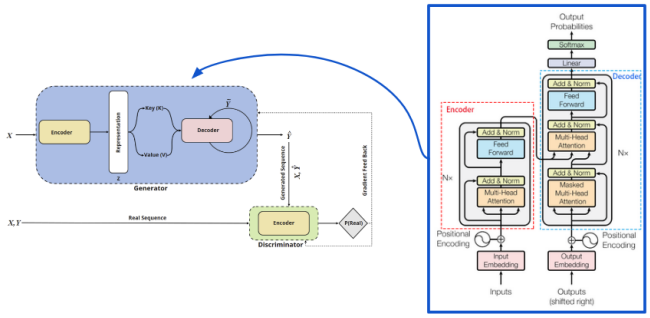

Figure 17. ClinicalGAN[19] model architecture.

A.6 StyleGAN2

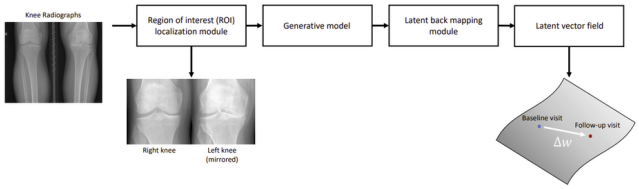

Figure 18. StyleGAN2[30] model architecture.

A.7 SADM

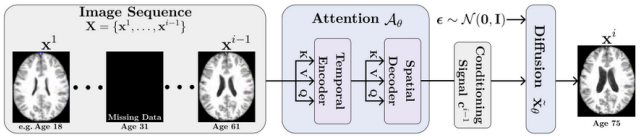

Figure 19. SADM[31] model architecture.

A.8 RVD

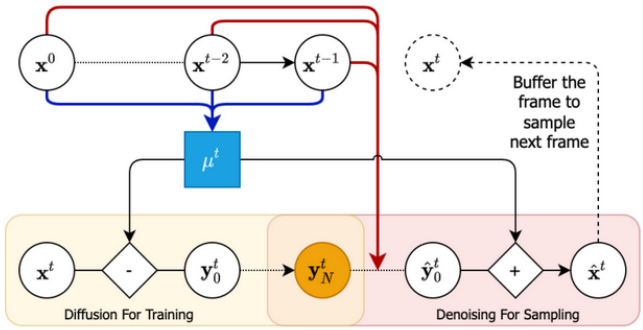

Figure 20. RVD[32] model architecture. Red arrow: ConvRNN predicts most likely next frame. Blue arrow: ConvRNN predicts context vector for denoision diffusion model

B Intuitive Baseline Results

B.1 ClinicalGAN Autoencoder

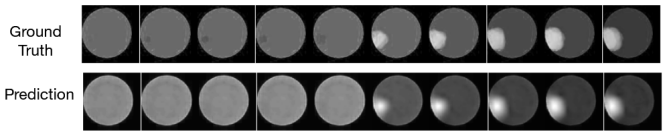

Figure 21. Reusult from ClinicalGAN[19] Autoencoder.

B.2 ClinicalGAN Transformer

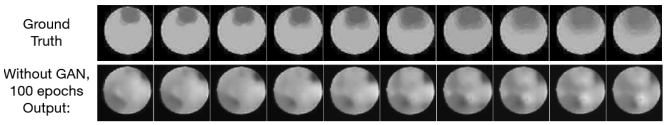

Figure 22. Reusult from ClinicalGAN[19] Transformer. Results are still not optimal.

B.3 VPTR Autoencoder

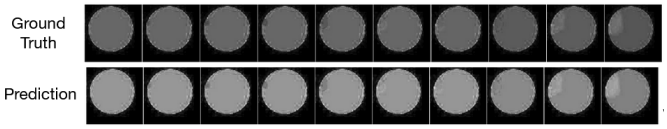

Figure 23. Reusult from VPTR[17] Autoencoder.

C Ablation Studies Tables

Note here the tables are too large to show, so the tables are in the next page.

Table 5. Ablation studies within VPTR[17] Autoencoder model using synthetic dataset 1.

Item	Input	Patients Number	Epochs/Training Time	Batch Size/Learning Rate	Training with/out GAN	MSE↓	PSNR↑	LPSIS↓	SSIM↑
1	Colored images	1000	100 epochs with pretrain/2.5 h on a RTX 2080Ti GPU	5/Fixed = 2e-4	With	Results too bad			
2	Gray images	1000	100 epochs with pretrain/2.5 h on a RTX 2080Ti GPU	5/Fixed = 2e-4	With	**2106.635**	**8.951**	**0.214**	**0.482**
3	Gray images	1000	100 epochs with pretrain/2.5 h on a RTX 2080Ti GPU	5/Fixed = 2e-4	With	2133.331	8.897	**0.214**	0.478
4	Gray images	10000	100 epochs with pretrain/8 h on a RTX 2080Ti GPU	5/Fixed = 2e-4	Without	2121.335	8.917	0.217	0.473

Table 6. Ablation studies within ClinicalGAN[19] Autoencoder model using synthetic dataset 1.

Item	Input	Patients Number	Epochs/Training Time	Batch Size/Learning Rate	Bottleneck	MSE↓	PSNR↑	LPSIS↓	SSIM↑
1	Colored images	1000	150 epochs without pretrain/2.5 h on a RTX 2080Ti GPU	5/Fixed = 2e-4	Average Pooling	Results too bad			
2	Gray images	1000	150 epochs without pretrain/2.5 h on a RTX 2080Ti GPU	5/Fixed = 2e-4	Average Pooling	Results too bad			
3	Gray images	10000	150 epochs without pretrain/74 h on a RTX 2080Ti GPU	5/Fixed = 2e-4	Average Pooling	2135.042	8.891	0.238	0.432
4	Gray images	10000	150 epochs without pretrain/19 h on a RTX 2080Ti GPU	5/Fixed = 2e-4	Linear Layer	**2120.449**	**8.919**	**0.219**	**0.444**

Table 7. Ablation studies within VPTR[17] Transformer model using synthetic dataset 1.

Item	Input	Patients Number	Epochs/Training Time	Batch Size/Learning Rate	Training with/out GAN	MSE↓	PSNR↑	LPSIS↓	SSIM↑
1	Colored images	1000	150 epochs without pretrain/13 h on a RTX 2080Ti GPU	1/Fixed = 1e-4	With	Results too bad			
2	Gray images	1000	150 epochs without pretrain/13 h on a RTX 2080Ti GPU	1/Fixed = 1e-4	With	1771.806	9.94	0.398	0237
3	Gray images	1000	150 epochs without pretrain/13 h on a RTX 2080Ti GPU	1/Fixed = 1e-4	Without	1636.922	10.038	0.411	0.185
4	Gray images	1000	150 epochs without pretrain/8 h on two RTX 2080Ti GPUs	2/Fixed = 2e-4	Without	1621.234	10.065	0.411	0.185
5	Gray images	10000	150 epochs without pretrain/140 h on a RTX 2080Ti GPU	2/Fixed = 2e-4	Without	**424.263**	**15.975**	**0.079**	**0.452**

Table 8. Ablation studies within ClinicalGAN[19] Transformer model using synthetic dataset 1.

Item	Input	Patients Number	Epochs/Training Time	Batch Size/Learning Rate	Training with/out GAN	MSE↓	PSNR↑	LPSIS↓	SSIM↑
1	Colored images	1000	150 epochs without pretrain/7.5 h on a RTX 2080Ti GPU	10/Fixed = 1e-4	With	Results too bad			
2	Gray images	1000	150 epochs without pretrain/7.5 h on a RTX 2080Ti GPU	10/Fixed = 1e-4	With	Results too bad			
3	Gray images	1000	150 epochs without pretrain/7 h on two RTX 2080Ti GPUs	10/Fixed = 1e-4	Without	Results too bad			
4	Gray images	10000	150 epochs without pretrain/25 h on two RTX 2080Ti GPUs	10/Fixed = 2e-4	Without	2367.657	8.465	0.323	0.331
5	Gray images	10000	60 epochs without pretrain/10 h on a RTX 2080Ti GPU	10/Fixed = 2e-4	Without	2689.788	7.897	**0.312**	**0.500**
6	Gray images	10000	60 epochs without pretrain/11 h on a RTX 2080Ti GPU	10/Fixed = 2e-4	With	**2648.101**	**7.953**	0.596	0.305