Gaze Synthesis for Online Communication

Sudipta Mondal
200856368
Supervisor: Dr. Miles Hansard
MSc Artificial Intelligence

***Abstract*—Gaze synthesis is an important area of research to help mimic face-to-face conversation in online communication, which is at its epitome due to current pandemic. This paper covers the study of gaze synthesis for online communication using convolutional neural networks with various loss functions, deeper layers, and optimizer. The inference is done on real time video frames captured via web camera, as well as static images. The experiments with the hyperparameters used in this paper reveal gaze synthesized by deep learning model with appropriate settings can aid in gaze correction of simpler eye structure (with minimal makeup/spectacles/without artificial eyelashes) quite well as compared to complex eye structure. Additionally, multi-scale neural network with spatial convolutions, perform better than uniform and deeper convolutions; and minimizing the objective function for structure of eyes and colours in the eye region are significant for gaze synthesis.**

***Index Terms*—Gaze synthesis, convolutional neural network, encoder**

I. Introduction

Eye contact is an influential element for human-to-human interaction that aids in understanding the other person's attentiveness, emotional state, cognitive state, and even assists in sentiment analysis. The orientation of gaze drives empathy and cooperation to form creative exchanges when creating music piece for instance [6]. Furthermore, due to the current pandemic situation, online communication has increased to carry out video conferencing for work, online education, interacting with family, and for online socialising. However, the lack of face-to-face communication and eye contact hampers communication and inter-personal relations. As far as technical perspective is concerned, when a person at the local side looks at the person on the remote window, it is perceived as looking downwards, a problem known as parallax [4].

During online training or academic sessions, such as, virtual class for singing, semantic synchronization problem arises, like coordinating breathing while singing [6]. Gaze synthesis can aid in synchronization by enhancing attentiveness of the cohort, as everyone can be made to look at each other. Psychological study mentions, gaze direction acts as a reliable indicator to analyse human emotional and mental states. For example, anger and happiness are related to direct gaze, whereas fear and sadness to averted gaze [1]. Therefore, gaze correction can promote positive communication with direct gaze. Other applications of gaze correction can be in images for group photos so that everyone's eyes are in the same direction, and in entertainment industry where the actors need to look at virtual characters or arbitrary direction when enacting and recording a scene, which is later substituted by visual effects.

With face-to-face communication understanding the other person might be easy, however with online communication this is quite challenging, especially as the task needs to be accomplished in real-time by a computer. Tracking the eye direction itself is a challenging problem given environmental conditions such as lighting, the instrument used, i.e., the camera and its complexity, calibration of the instrument such as the focal length and resolution of the device being used. In the past hardware-based approaches were used to correct the gaze after tracking eye direction, requiring powerful cameras or display monitors, which is not feasible for general public to use, and is often expensive. But, with hardware advancement of computationally powerful inbuilt cameras in devices, and technological advent of machine learning and deep learning models, along with extensive amount of data availability, it is now possible to build applications that can be used with embedded web cameras. These software-based approaches focus majorly on learning from the images and altering the gaze correction in real time.

A. Contributions

Gaze redirection is an ongoing research area aiming to rectify the eye gaze in real time. In this paper, the aim is to study about a gaze synthesis and correction approach. The contributions are:

- Comparison of different loss combinations, optimizer, and batch sizes for training a gaze synthesis model.
- Incorporate focal length calibration in real time, used for the gaze correction.
- Compare effect of VGGish deeper and uniform convolution layer.
- The experiments have been logged and recorded with live monitoring tool, Wandb.
- Study the effect of gaze estimation with one of the pretrained model, which is trained on unconstrained images and employs long-short term memory (LSTM) neural network.

B. Paper Organization

The paper is organized into sections. Section II gives an overview of related work for gaze correction, Section III details the research focus, Section IV covers datasets used for the study, and implementation details, followed by Section V covering evaluation of the experiments performed. The final sections, Section VI, VII, and VIII, ends the paper with discussion, future work and conclusion, respectively.

II. Related Work

A. Deep Warp

Deep Warp [3] comprises of deep convolutional neural network architecture trained end-to-end for coarse-to-fine (CFW) image processing, image warping and intensity correction by lightness correction module (LCM) performing pixel-wise correction of brightness for photo-realism of the image. As mentioned in the paper [3], the earlier image synthesis methods used encoder-decoder architectures to learn internal representations of the image pixels, followed by transformation. However, these methods suffered from regression-to-mean effect, i.e., bias towards mean of the training data. Additionally, some of these methods relied on a specific angle of gaze direction during training, without consideration to other angles that can occur in real world scenarios. For instance, the notion of gaze redirection with supervised learning as advised in [11], employed random forests (a machine learning algorithm) to predict the warping fields based on the specific angle. DeepWarp model, Figure 1, allows redirection angle to be specified as input which can change continuously in certain range, instead of a fixed angle. It employs multi-scale neural network (extraction of features at different scales) to learn flow field (warping field) instead of low-dimensional parameters, with the help of prior knowledge of the gaze correction angular offset to redirect the input image. The inputs to the model are angular offset, anchors (feature points of the eyes), and the input image. The model is a deep neural network consisting of convolutional layers with batch normalization, and ReLU for non-linearity activation and fully connected layers with same-mode convolutions (dimension of input and output are same). The CFW module consists of a coarse warping ($0.5\times$ scale network) module that produces pixel-flow field, followed by upsampling input image with bilinear sampler (method that uses linear interpolation to compute a pixel's value by referencing the nearby pixels). After this, fine warping module ($1\times$ scale network) takes the coarse warped image, the upsampled coarse image and the input data (image, embedded angle vector and embedded feature points), to generate the output via bilinear sampler. Finally, after the warping is complete, lightness correction module takes coarse and fine features, in addition to the input features to perform element-wise transformation to blend the colours of warped pixels. A dataset was created specifically for this network to capture certain gaze direction as ground truth and vertical gaze as original image.

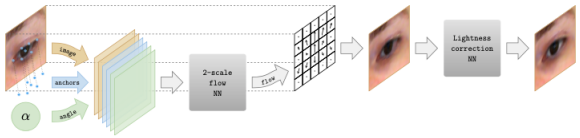


Fig. 1. DeepWarp model [3] (left to right flow): the model takes eye image, anchors (feature points of the eyes), and target correction angle as input. These inputs are then fed to the multi-scale deep convolutional neural network to determine the flow field of the pixels. The flow field is applied to the original input eye image for warping. The warped image is passed over to another neural network for light correction and synthesize corrected eye gaze image as final output.

DeepWarp model being a pure warp-based approach, suffers from inability to work with occluded eye images, such as reflection due to spectacles in the image that might occlude the eye completely, and secondly, the range of gaze redirection depends largely on the training data.

B. GazeDirector

GazeDirector [2] relies on recovering appearance of eye region in 3D, to generate eyeball. This model tries to mimic real life eye movement by considering the eyeball rotation separate to the eyelid. It does not rely on dataset for learning, instead focuses on optical flow fields derived by the model, using GPU rasterization (a process of creating 3D structure from a surface of

object, represented as virtual polygons or triangles with vertices, with each vertex containing information about colour and texture; followed by converting each triangle of 3D model into pixels on 2D screen). The framework is shown in Figure 2. The first step in this approach

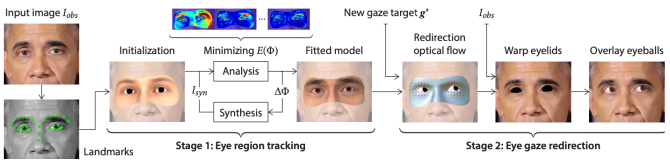


Fig. 2. GazeDirector framework [2]: in this framework the observed image is the input, where landmarks from the face is identified to initialize the model parameters. In the stage 1, the model objective is to analyse and synthesize the original image, by minimizing the reconstruction energy between the original and synthesized image. This leads to finding the optimal parameters for synthesizing original image. In stage 2, the learned parameters are modified in accordance to the target gaze angle; followed by overlaying the corrected eyeball on the input image.

is to synthesize eye region by analysing the sparse facial landmarks as input, namely: eyebrows, nose, and eyelids. It aims to learn parameters describing geometric shape of eye region, texture, position, and illumination. Once the parameters are configured, rasterization is performed. Here, the eye region model learns by minimizing reconstruction energy, a weighted sum of data terms (photometric error of image pixels and sparse landmark similarity), and prior terms (penalizing unlikely facial shape and texture, and mismatched eyeball gaze direction and eyelid position). Photometric error is minimized by comparing the pixel-wise difference between the synthesized image and original observed image, averaged on foreground pixels of the 3D model.

The second stage of the model involves gaze redirection once the parameters are fit to the model in the first stage. The parameters are then modified to form the new gaze direction. The eyelids in the original image are warped with flow field derived by the model; followed by redirecting eyeballs, and then combining warped eyelid and synthesized eyeball back into the original image.

The drawback of the model is it is incapable to synthesize gaze for occluded eye region. To illustrate further, if a person is wearing spectacles during online communication, this method cannot correct gaze in this scenario because it might warp the spectacles which will be part of the eye region. Secondly, it does not consider other facial expressions like blinking. Therefore, this algorithm is not suitable for real-time [5] applications.

C. Correcting Eye Gaze

The work done by Chihfansu et al. [4] aimed to correct the missing eye contact in online communication with the motivation from the DeepWarp model to rectify gaze in video frames before sending it to the remote side of the network. The novelty in their work includes estimating eyeball rotation angle based on camera position and participant's eyes; addition of dense blocks to improve feature utilization in hidden layers, and introduction of loss functions to preserve eyeball and eyelid shape that focuses on brightness and flow of pixel learnt by the network to reduce colour change in peripheral regions of the warped eye. The framework is shown in Figure 3.

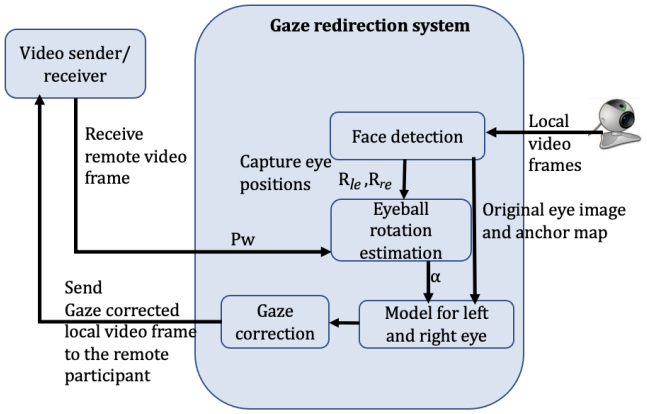


Fig. 3. Correcting Eye Gaze for online communication [4]: (P_w) is the assumed remote participant on screen (determined by retrieving the centre of the remote participant window), (R_{le}) and (R_{re}); α is the estimated angle rotation computed with the help of local participant's eye and remote participant's assume location (P_w). The gaze correction module is fed with the trained models, input frame and target eyeball shifting to generate frame with corrected eye gaze. Finally, the corrected video frame is sent to the remote participant

The objective of the gaze redirection system, is to modify the gaze direction from P_eP_w to P_eP_c by rotating the eyeball in x and y coordinate axes, shown in Figure 4, where P_e, P_c and P_w denote the centre of local participant's eye (computed with the help of focal length, interpupillary distance and remote screen coordinates), centre of camera (determined from specifications) and centre of remote participant's eyes on screen. The angles are determined by computing the difference between the coordinate positions along x-axis and y-axis separately and dividing by the difference of positions in z-axis. This is followed by applying inverse tangent to get the degrees of each fraction separately, then sum the angles to obtain the final eyeball rotation angle, [4]. Once the eyeball

rotation angles are estimated, eye image is warped based on the rotation to overlay on the original video frame.

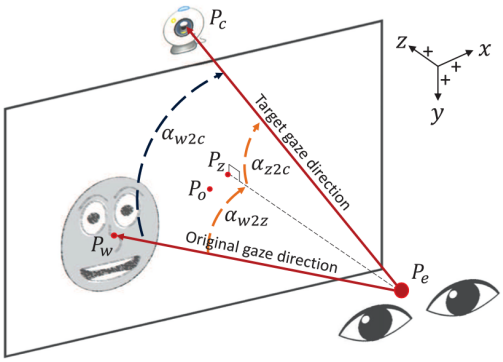


Fig. 4. Eye Gaze Redirection [4]: a coordinate system with x,y,z axis is considered. (P_e) is the computed local participant's eye centre, (P_w) is the assumed remote participant on screen (calculated by retrieving the centre of the remote participant window), (P_c) is the camera position, and (P_z) is the centre of the physical screen. The eye gaze redirection aims to correct gaze from the original gaze direction (P_w) to the camera (P_c), by rotating the eyeball with the estimated eyeball rotation angle α.

Facial landmark is used to get the eye region from image, and corresponding anchor map. The eye image, anchor map and eyeball rotation angle are fed as input to the warping model to generate warped eye image by minimizing the objective function that measures the distance between the warped image and the ground truth. The network consists of (i) an encoder that transforms input 2D eyeball rotation angle to 16D vector to form a feature map, (ii) a warping network with input as the segmented eye image, encoded feature map of rotation angle, anchor map, and outputs pixel flow field for image in addition to a feature map for colour correction, and (iii) colour network similar to lightness adjustment module in DeepWarp to rectify brightness field and interpolate between white pixels and original pixel colours to maintain the photo-realism of the synthesized gaze.

As mentioned in the paper [4], this method is faster than the DeepWarp because it has one-level warping model, and does not perform bilinear interpolation twice, thereby reducing the computation cost to half; and claims to work well even with occlusion of eye region, as long as the eye ball is clearly visible.

The drawback of this method is, it assumes that the participant is looking at the centre of the remote participants' screen during the online communication and corrects eye gaze by relative transformation. Therefore, this method will not work if the participant is looking somewhere else instead of the remote participant(s) screen in reality.

D. Correction using deep neural network

Another contribution in the field of Gaze correction is ECC-Net model [5] which aims to redirect the gaze from arbitrary direction to the centre without any need of redirection angle; by using a deep convolutional neural network to generate a vector field, and brightness map to correct gaze. The authors trained the model on perfectly labelled synthesized image, and a novel dataset for training was created using UnityEyes user interface, where gaze can be moved by moving a cursor. Another dataset was created with proprietary facial landmark detector developed by Intel. The input to the ECC-net is a concatenation of cropped eye image, and a redirection target angle. The input is fed to a series of convolutional layers, followed by a series of deconvolutional layers, and the output is: (i) a vector field to warp input image, and (ii) a brightness map passed through sigmoid non-linear activation for brightness adjustment. The input target angle is set to zero during inference, as the goal is to move the gaze to the centre.

The convolutional blocks comprise of three depth wise-separable convolutional layers with a residual connection that skips the middle layer. Depth wise separable convolutions are type of convolutions that work on the depth of the filters (each channel), followed by spatial convolution (1x1) to capture important features from the image and reducing number of parameters at the same time.

The authors performed bidirectional training, where first direction tries to minimize the loss between model's gaze correction and the ground truth (note: in the training dataset for each eye, two pairs are present, one looking at random direction which forms input and other looking at the centre forming the ground truth). The second direction has the reverse task of reconstructing the original image from the gaze corrected output image and the input angle.

The bidirectional training claims to have reduced artifacts and resulting gaze direction is meant to be more natural. It has been mentioned that this model can also do gaze prediction without any additional training. However, the drawback in this approach is the need of perfectly labelled ground truth images in the dataset.

E. Overview on Gaze estimation

Gaze estimation is a related computer vision task to predict the angle of human eye gaze, which can be amalgamated with gaze correction mechanisms. Traditional methods involved creating geometrical representation of eye region to capture the relation between pupil

and iris for estimating gaze direction, by fitting the elliptical structure of iris when the eye moves. These were categorised into intrusive (requires devices attached to the body) and non-intrusive (does not cause any intrusion) techniques, [20]. However, with the advent of machine learning, convolutional neural networks are used to extract the features from the eye region image and feed the extracted features to fully connected layers to generate the gaze vector. These models aim to learn direct mapping between direction of gaze to eyes or facial appearance. For the training purposes, the loss functions like L2 norm is used, for minimizing the distance between the ground truth and predicted gaze vector, [14]. Even though most of these methods rely on convolutional neural networks for learning features, recent algorithms are based on bidirectional LSTM models [15]. Bidirectional LSTM is a recurrent neural network based deep learning model which can learn from the past and future sequences of the data. Attention based transformer models to learn features are still an exploration area for gaze estimation, [21].

As the methods and algorithms for gaze estimation are emerging, new datasets are also getting generated to aid these methods. For instance, CAVE dataset [7] consists of images captured in constrained environment and annotated with eye gaze angle and head pose, MPI-IGaze dataset [20] was created with images from laptop users, Gaze360 [15] is a dataset created in unconstrained environment using mobile devices. Although, most of the models require robust and large datasets, some models use few-shot adaptive gaze estimation [13], that aims to estimate gaze using only few examples.

III. Gaze synthesis for online communication

For the study of gaze synthesis and correction in this paper, the motivation is [4] where the warping-based model is used. In this work, experiments have been performed with different batch size, stochastic gradient as optimizer in order to compare with the Adam optimizer implementation in the base model. In the previous work, the focal length is calibrated by determining the corners of the eyes, as shown in Figure 5; in isolation, by computing Euclidean distance between left and right eyes, multiplied by an approximate distance of person from the computer (50 cm), then divided by the interpupillary distance of 6.3 cm. However, in this paper the focal length calibration has been integrated along with the gaze correction system to avoid an additional step.

Fig. 5. Focal length calibration example, focal length:1055mm. The focal length is computed by first detecting the facial landmarks followed by computing square root of the squared distance multiplied by distance from screen, divided by the interpupillary distance of 6.3

Furthermore, to analyse the effect of deeper and uniform convolutions, additional convolutional layer has been incorporated in the warping network following VGGish pattern, with uniform kernel size of 3x3, with same padding [15]. The motivation was considered from image classification task in [15], as uniform structure has a few benefits: (a) with more convolutional layers better non-linearity is implemented which makes the decision function to be more discriminative, and (b) decreases parameters. For instance, when comparing one 7x7 conv. layer to three 3x3 conv. layers, the number of parameters reduces from 1x7x7=49 to 3x3x3=27, reducing the number of parameters by 45%, while still effective in capturing information from the image. With fewer number of parameters it is believed that the model supposedly converges faster.

The loss functions have been studied to analyse the effects of gaze correction with various combinations of the loss function. The original loss function consists of L2 loss to measure the distance between warped image and ground truth, shape loss to determine the loss of pixels in reconstructing the eyeball and eyelid, colour based loss to reduce visual artifacts and reduce colour adjustment that might get introduced during model training, and lightness control loss to make the colour adjustment smooth and improve naturalness of the synthesized eye.

An attempt has been made to perform gaze prediction before sending local participant's video to the remote participant during online communication, with pretrained model for transfer learning. The source domain is trained with wide range of gaze and robust dataset, with LSTM model, Gaze360 [15]. The aim was to avoid the assumption that a subject is looking at the centre of remote window and capture the gaze angle. As mentioned in the paper [15] this is the largest publicly available dataset

comprising environmental settings of both indoor and outdoor with wide ranges of head pose. The key to this Gaze360 dataset is unconstrained collection of eye gaze in varied lighting environments regardless of blur in the subject, wide range of head pose angles, even with occlusion due to large head yaws and angles. Furthermore, the volunteers were of varied ages, with nearly equal percentages of male and female participated for the collection of the images.

IV. Methodology

A. Dataset

As a general method to collect images of eye gaze, the datasets are created in constrained environment and human participants are asked to look at certain point facing towards the camera at some distance, in order to capture the head pose along with the eye gaze, [4], [7]. In these datasets when the participant looks at the marking, it forms the ground truth image and looking arbitrarily at any other point forms the input image for the model. In most cases frontal face is used as a data sample and any images where eyes are closed are discarded. In some instances crowd-sourcing is used to capture the eye gaze, where smartphones or integrated camera are utilized to collect data for unconstrained head movement, [15].

The dataset used for the study are DIRL and CAVE dataset with details in the following sub-sections. As mentioned in the papers [4], [7] datasets were created with constrained lighting environment and head movement. A difference noticed during the analysis was, DIRL consists of cropped eye region images, whereas CAVE consists of images with full face. Therefore, the datasets were first pre-processed to create training data for each left and right eye. The pre-processing is done such that the eye region's width is considered 1.5 times the eye length along with some fraction of aspect ratio to consider the eye lids. 14 anchor points (7 for each eye) are used to create the anchor map and centre of eye. Finally for each sample data, a dictionary is created consisting of original image, anchor map, vertical and horizontal eye angles obtained from the image file name, anchor point for corners of eye, original image size and aspect ratio adjusted with image of width 64 and height 48 pixels.

a) DIRL: DIRL gaze dataset contains 16,196 images with 1920x1080 pixels resolution with dataset collected from 37 Asian male and female volunteers inclusive of some volunteers with spectacles, [4]. The images captured consists of mainly 5 head poses with angles of 0°, ±10°, and ±20°. DIRL has another dataset in text format where for each eye image, the labelling consists of 0 for right eye and 1 for left eye, followed by anchor points from 0 to 7 (last one being the pupil) representing the facial landmark of two eyes; each anchor point has x-axis and y-axis coordinate.

b) CAVE: The Columbia Gaze dataset is a publicly available dataset as the benchmark for gaze tracking [7]; with 5,880 high-resolution images of 56 ethnically diverse and varied age people (32 male, 24 female). For each participant, images for combination of five horizontal head poses (0°, ±15°, ±30°), seven horizontal gaze directions (0°, ±5°, ±10°, ±15°), and three vertical gaze directions (0°, ±10°) is captured. CAVE dataset consists of only facial images. Therefore, to generate a labelled dataset like DIRL, pretrained haarcascade classifier is used to determine the frontal faces in the image along with dlib and OpenCV libraries, to get the coordinates for each anchor point of eye and pupil and generate the labelled dataset. Haar cascade classifier was built by Viola et al [22] as the machine learning algorithm for face detection from image and video. Their work involved ensemble learning with Adaboost to determine the most important Haar features (like convolutions) by cascading classifiers which combines weak learners to build a strong learner for the detection task.

B. Implementation

The implementation has been adapted from the publicly available open-source gaze correction system [4] with higher version of Python 3.7 and Tensorflow version 2.0 for the model training. Wandb API is used for monitoring, and to capture logs and losses of the experiments.

The convolutional layers are initialized with the Kaiming Initialization, or He Initialization. It is an initialization method for neural networks that takes into account the non-linearity of activation functions, such as ReLU activations [17].

While generating training data, the eyes and eye lids are first cropped and separated by creating eye and lid masks with the help of the eye anchor map in the dataset; so that during the training the focus is on eyes and the eye lids are intact. Furthermore, image pair list is created for each combination of the angle considering source and target. For instance, if there are 98 images, then there will be 9604 (98 x 98) training pairs. These training pairs are further split into two batches: first batch consists of training pairs where the difference between horizontal and vertical gaze angle is less than a certain threshold forming easy batch training data, and the second batch

consists of rest of the training pairs which exceed the threshold angle limit, which is harder to train.

In order to test the trained model for real-time online communication, Windows laptop is used to simulate local and remote participant window with python socket API. The image resolution is set to 640x480 like the original work to avoid latency. The face is detected using the dlib open-source library for getting the 68 facial landmarks from the frontal face of the application window in real time. Python multiprocessing package is used for parallelism of face capture in one thread and perform gaze correction with another thread.

The model architecture, Figure 6, consists of encoder with fully connected dense layers, receiving input as vertical and horizontal angles and output generated is an angle map. The anchor map, angle map and input image is concatenated and then average pooled, followed by downscaling, to extract coarse features. The resized coarse features along with concatenated input is fed to another network to extract finer features and generate a flow field. This flow field is applied to original image and the fine features are used for light correction. Finally, the warped image with lightness corrected is synthesized.

For implementing the Gaze360 pretrained model for gaze estimation, the model and the publicly available python modules is used. The video frame from which the gaze is to be estimated, along with the bounded box of the face is resized to 224x224 with RGB channel and fed as input to the pretrained model. The gaze angle is applied to rotate the original gaze direction to the centre of the remote participant's window, with sine and cosine of the estimated gaze angles.

C. Training

The training is performed with smaller batch sizes in 2 GPUs. The different sets of GPUs used were GeForce RTX 2070 and Quadro RTX 6000. The number of epochs chosen for all the experiments was 30 epochs with an initial learning rate of 0.001 for comparison. The datasets were split into 70:30 train and validation set to save the best model after evaluation on the validation set. Data iterator is used to shuffle the data and create small batches for each iteration. If the validation loss is less than the training loss, then the best model at that point is saved. However, if the validation losses are greater than the training losses, then the learning rate is decreased by 0.9 times the learning rate; and further learning is stopped if there is no improvement for the next 16 steps. The models are trained for both left and right eyes, separately.

The models compared for training eye gaze correction with the datasets are: base model with Adam optimizer with a batch size of 64, model with smaller batch size of 32, model with SGD optimizer and a batch size of 32, model with additional VGGish convolutional layers, and models with different combination of loss functions.

V. Experimental Results

A. Evaluation of Gaze correction

The comparison of different models for correcting gaze on the left eye for the DIRL dataset is shown in Figure 7 and Figure 8. The model with stochastic gradient descent optimizer stops learning early and ends up with higher ranges of L2 loss (difference between ground truth and synthesized image) and total loss (L2 + shape + colour loss, to capture structural and colour difference between the eyeball, pupil, iris, sclera, and eyelids of ground truth and synthesized image); demonstrating dissimilarity between the synthesized gaze and ground truth, Figure 8. On the other hand, base model with the Adam optimizer continues learning and converges to minimize loss to synthesize images quite close to the ground truth image. The base model with a batch size of 64 synthesizes eye that is visually appealing and closer to the ground truth. Similar results are obtained with batch size of 32, with faster convergence. The VGGish model with the DIRL dataset synthesizes the corrected gaze with additional artefacts, probably because of the deeper layer and uniform convolutions.

The loss curve for model trained with CAVE dataset demonstrates lower losses with SGD optimizer with synthesized image perceptually similar to ground truth. However, it shows dissimilarity between the ground truth and synthesized image with added artefacts, when VGGish convolution is used, Figure 9. The dissimilarity to ground truth can be due to smaller dataset size for the model to learn from. The image synthesized with both batch size of 32 and 64 are nearly similar, however the loss curve demonstrated batch size of 64 converges faster, unlike the DIRL dataset. Another observation is that the learning rate (LR) for SGD optimizer drops to 0.0007, whereas, for the other models LR decreases till 0.0006, showing a tendency to slow learning as compared to other models.

To explore the behaviour of the deeper model with the CAVE dataset, the last convolutional layer was changed to spatial convolution (1x1 filter), and it was observed that artefacts get added. But it can be deduced that a uniform convolution is not beneficial for gaze synthesis. Rather, a multi-scale network learns the important

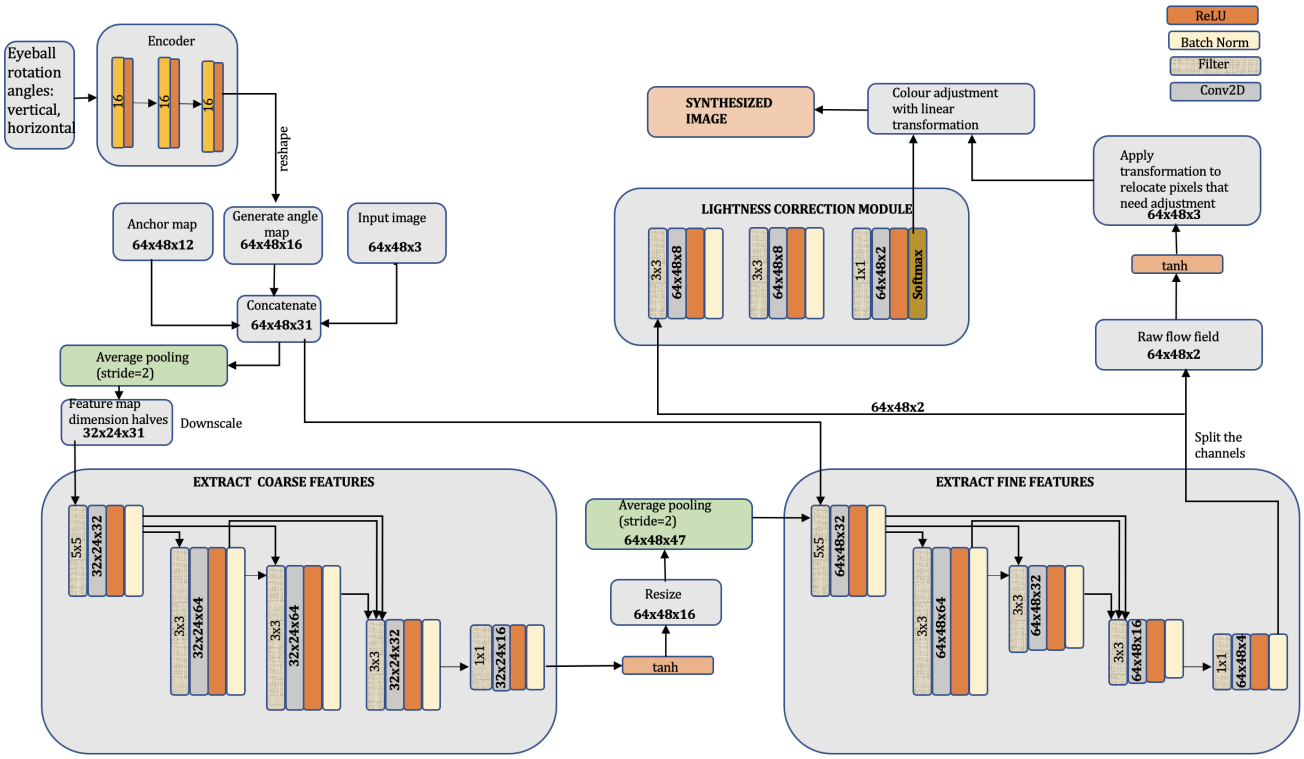


Fig. 6. Model architecture consists of encoder, deep convolutional network for extracting coarse and fine features, lightness correction module for colour correction of the synthesized image.

features from the latent space, and efficient in gaze synthesis.

To further study behaviour of the model, evaluation with different loss function combinations is performed, refer to Appendix A Figure 14 and Figure 15. For the ease of understanding, the acronyms used are: (i) L2C for distance and colour loss, (ii) L2EB for distance and eyeball loss, (iii) L2EL for distance and eyelid loss, (iv) L2P for distance and colour smoothing, (v) L2S for distance and shape loss (loss for correcting both eyeball and eyelid), (vi) L2SC for distance, shape, and colour loss, and L2 for only distance loss. With L2C as objective function, the colour difference between the iris, pupil and sclera of the ground truth and synthesized image is reduced, but the colour difference in the edges is prominent. However, it is clearly noticed that the network does not learn the shapes, because the eyeball gets smudged, eyelid disappears at times when there is reflection in the original image. L2 loss stops learning early without considering the shape and colour of the eyes. L2EB as loss function shows differences in the colour of the lower and upper lids, and in some cases the lower eyelid blends with the eyeball. The distortion in the eye images is even increased further if L2EL is the objective function, and the eyeball is distorted. On the other hand, with L2S, disappearance of eye lash is noticed in addition to slight smudging of the eyeball. With L2SC as loss function, the shape of the eyes and the colour adjustment is appropriate. However, when comparing the overall eye region image of the synthesized image with the ground truth, it is noticeable that the edges of the image and eyeball have colour differences.

With regards to training, Figure 10, the model stops learning earlier with L2, L2C and L2P losses as compared to L2EB and L2EL. L2S continues learning and L2SC converges early. The Table I and II shows the loss function with different models, and it is evident that as the loss function becomes complex the loss values are also in higher ranges, as the model tries to synthesize gaze by minimizing the combination of loss functions. As per the observation of the learning rate curve, for L2SC the LR decreases to nearly 0.0055, L2C to 0.0065, and rest of the models stopped at a higher learning rate of nearly 0.0078, suggesting L2SC model takes longer time to learn.

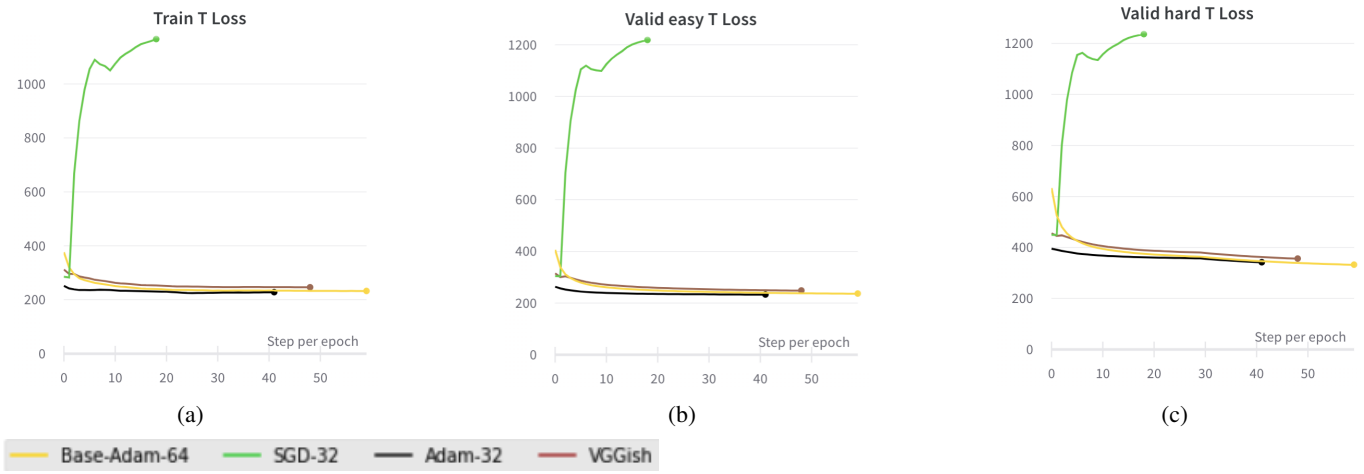
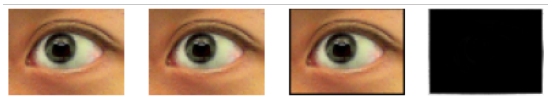
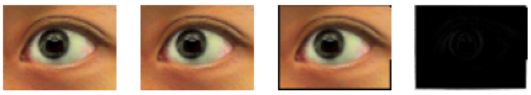


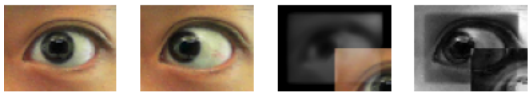
Fig. 7. Comparison of training and validation total losses for DIRL dataset: the total loss with SGD optimizer is very high and the learning stops early. VGGish network with Adam optimizer stops early as well. The base model with batch size of 64 continues learning, whereas batch size of 32 converges early.



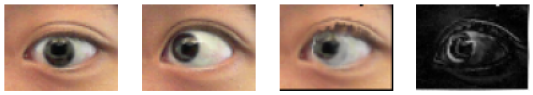
(a) Adam+batch size 32: difference between ground truth and synthesized image is very small.



(b) Adam+batch size 64: difference between ground truth and synthesized image is very small.

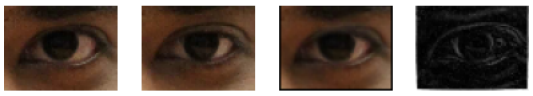


(c) SGD+batch size 32: synthesized image underfits the training data. The overlap between the ground truth and synthesized image is quite large.

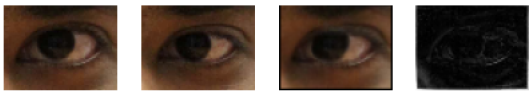


(d) VGGish deeper convolution: gaze corrected with additional artefacts

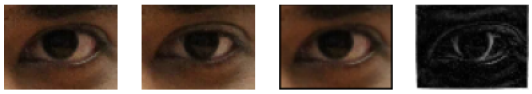
Fig. 8. Comparison of eye gaze correction for DIRL dataset. From left to right: input image, ground truth, synthesized image, difference between the ground truth and synthesized image



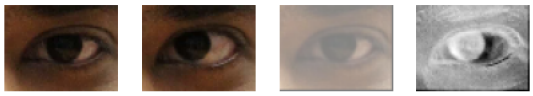
(a) Adam optimizer + batch size 32: very less difference is observed between the ground truth and the synthesized image



(b) Adam optimizer + batch size 64: very less difference is observed between the ground truth and the synthesized image



(c) SGD optimizer + batch size 32: perceptually looks closer to input image

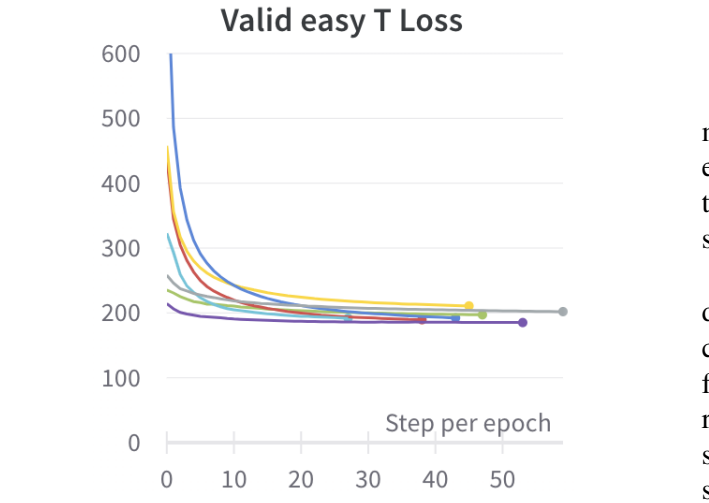


(d) VGGish deeper convolution: additional artefacts with complete dissimilarity with ground truth

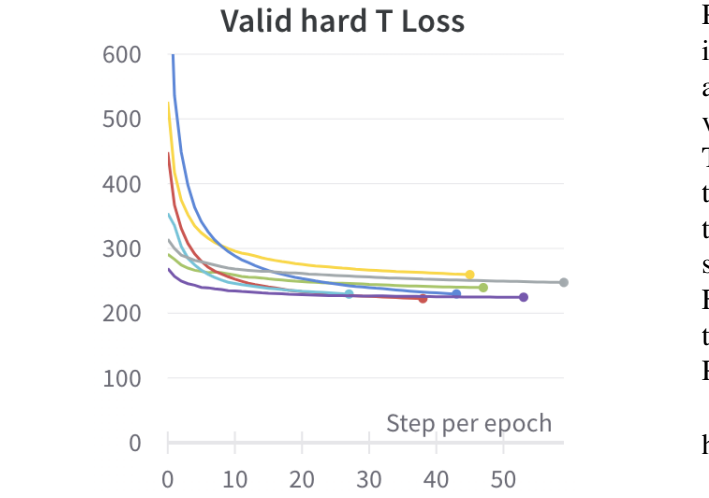
Fig. 9. Comparison of eye gaze correction for CAVE dataset. From left to right: input image, ground truth, synthesized image, difference between the ground truth and synthesized image. Except for the Adam optimizer with 64 and 32 as batch size, rest show significant mismatch between the ground truth and generated gaze image.

Additionally, to study the effect of losses of the models trained with DIRL dataset, the inference is done on a simple frontal face, and complex image of some actors picked from NewGaze dataset [8] which was originally created for unsupervised gaze correction using generative

(a) L2P and L2C losses are higher in the beginning and stops training early. When considering shape the training loss starts with lower losses and continues training longer.



(b) The validation loss of samples within the threshold of eye movement shows similar trends as the training loss



(c) During the validation with hard samples the losses are higher in the beginning as compared to the training

Fig. 10. Loss analysis

TABLE I
LOSS COMPARISON WITH CAVE DATASET FOR LEFT EYE

Model	Training loss		Valid easy loss		Valid hard loss	
	L2	Total	L2	Total	L2	Total
L2C	203.9	204.5	178.6	179.2	207.4	208.3
L2EB	200.9	206.2	178.7	183.6	211.8	220.9
L2EL	212.5	223.5	181.5	191.9	215.3	229.7
L2P	211.6	212.1	177.3	177.8	210.6	211.2
L2S	202.8	216.9	186.9	199.6	220.5	240.5
L2	217.4	217.4	181.5	181.5	212.8	240.5
L2SC	245.1	255.6	200.4	210.8	239.7	255.8

TABLE II
LOSS COMPARISON WITH DIRL DATASET FOR LEFT EYE

Model	Training loss		Valid easy loss		Valid hard loss	
	L2	Total	L2	Total	L2	Total
L2C	195.2	196.1	206.3	206.8	263.1	256.2
L2EB	190.3	196.8	205.6	211.5	265.5	266.7
L2EL	199.1	211.4	205.9	216.7	259.5	278.4
L2P	200.9	203.3	203.8	204.3	254.7	258.2
L2S	207.5	227.6	209.3	227.0	265.5	294.6
L2	202.9	202.9	205.2	205.2	302.3	302.3
L2SC	197.7	219.7	211.2	230.4	304.8	270.0

model and consists of wild images in unconstrained environment. To perform this experiment, position of the web camera is approximated in the configuration for synthesizing centre gaze.

The gaze synthesized for the frontal face looking downwards to bring it to the centre, Figure 11, without considering the shape of eyeball and eyelid in loss function, takes into account the colour around the eye region, but the shape gets distorted. Considering the shape in loss function generates left eye effectively with slight distortion in the right eye gaze synthesis. To perform a quantitative measure of the ground truth and the reconstructed gaze corrected image, an evaluation with peak-signal-to-noise ratio (PSNR) is carried out. PSNR is an evaluation metric for the assessment of the image quality for reconstruction or restoration of images as compared to the original image. Higher the PSNR value better the resolution of the reconstructed image. The observations, as shown in Table III, and Figure 12 that captures PSNR with each epoch during the model training, PSNR of the model that considers distance, shape and colour loss is the highest for both the datasets. For the DIRL dataset PSNR for model with L2 loss is the lowest, and for CAVE dataset L2EB shows the lowest PSNR.

Furthermore, when studying the effect of the model on harder samples with spectacles, Figure 16, a slight align-

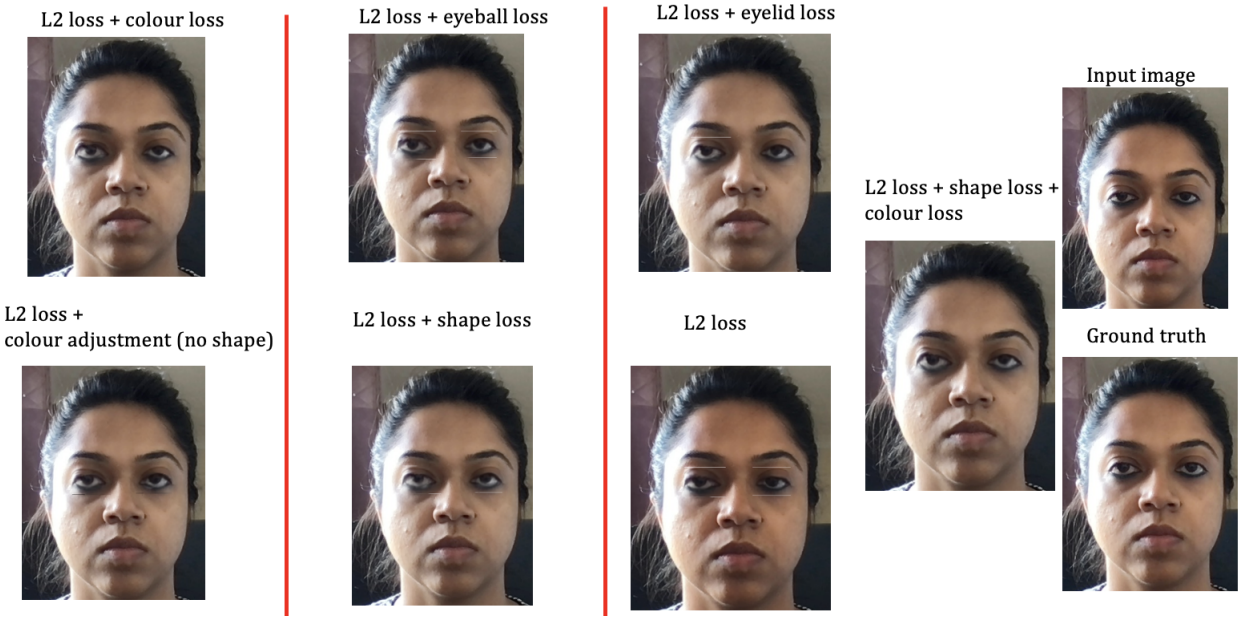
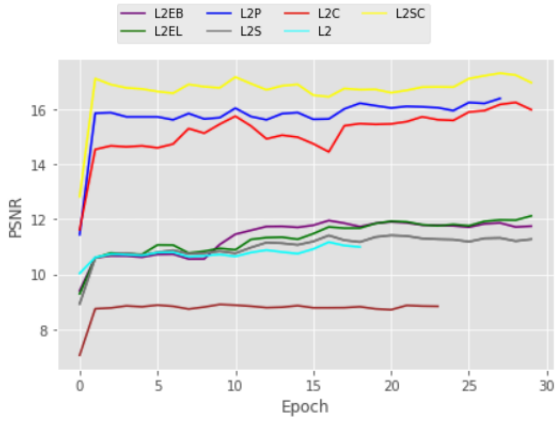


Fig. 11. Frontal face looking towards left side corrected to gaze at the centre of the screen. L2+colour loss and L2 +colour adjustment, does not consider the shape of eye. With L2+eyeball, eyelid is smudged. With L2+eyelid, eyeball, and eyelid overlap. L2+shape and L2+shape+colour results consider the shape. At inference L2 produces poor result.



(a)

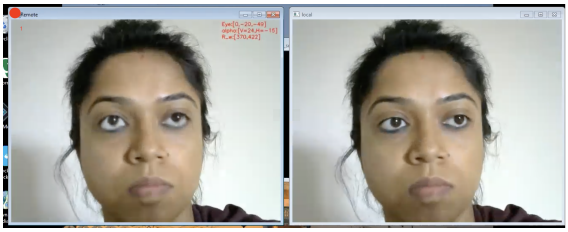
Fig. 12. L2SC has the highest PSNR, followed by L2P, L2C. L2S, L2EB, L2EL and L2 are in a range lower range, with VGGish having the lowest PSNR.

ment distortion is observed in the synthesized images, though the eyeball is centered. A more complex eye with eyeball vertically and horizontally angled to the corner, long eyelashes, and eye region partially covered by hair, synthesizing gaze is demonstrated as a challenging task to achieve, Figure 17.

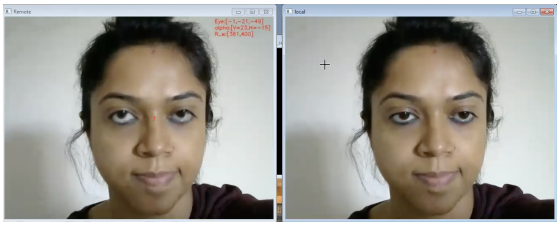
B. Evaluation of Gaze correction in real time streaming

For the study and comparison of the gaze correction in real time video for online communication, the first inference was performed with pretrained weights on Windows device without CUDA enablement, i.e., inference performed on CPU based device. To capture the video frames, basic external HD webcam with 30 fps (frames per second) was employed. Gaze synthesis of two captured frames is shown in Figure 13. It is demonstrated that when the gaze direction is at the centre of the remote window, the synthesized gaze transmitted to the remote window is centred. However, when the gaze direction is arbitrary other than the centre, synthesized gaze is directed upwards. This is probably because the gaze angle of the local participant is not estimated, rather an assumption is made that the participant is looking into the centre of the camera. The second observation is the right eye has a cross eye effect with slight smudging on the eyeball and sclera. The possible reason could be lack of enough right eye training data.

The subjective evaluation of gaze correction in collaboration with the estimated gaze angle, with transfer learning of Gaze360 model trained on unconstrained images, rectifies the gaze angle to camera centre in some instances (when looking at arbitrary direction). Although the results are not quite remarkable, and it takes longer time for inference in CPU based device. This can be probably resolved by retraining the Gaze360 model gray scale, along with accurate head pose estimation to avoid large eyeball rotation. Currently, a bounding box is



(a) Right: local participant gazing at the top left corner of the remote participants window. Left: synthesized gaze in the remote window.



(b) Right: local participant gazing at the centre of the remote participants window. Left: synthesized gaze in the remote window.

Fig. 13. Simulation of gaze redirection in real time: the local participant gazes in the remote window. When gaze redirection is triggered, the video frame is modified and then transmitted to the remote window.

passed to the model as estimated head position.

VI. Discussion

The experiments and evaluation demonstrate that the performance of the gaze synthesis models trained with DIRL dataset is better than the CAVE dataset, probably due to the dataset size, which allows the model to learn better with more data. Adam optimizer is superior to the stochastic gradient descent optimizer for gaze synthesis. The VGGish model trained with DIRL dataset is better than training with CAVE dataset, although artefacts do get added. However, inference of the VGGish network demonstrates uniform network architecture is inefficient in gaze synthesis. The comparison of the loss functions illustrates that L2 loss to compare the ground truth and synthesized image, in combination with the colour loss and shape loss achieves higher image quality. The training with batch size of 64 appears consistent performance for both the datasets.

The inference of the models on static images to centre the gaze, demonstrate that gaze synthesis of simple eye structure is better when compared to complex eye structure like occlusion with hair, or even higher degree of vertical and horizontal gaze. The inference on the online video with the trained models is comparable to the inference on static images. The gaze corrected before transmitting the local participant's video frames to the remote participant's window demonstrate that when the participant looks at the centre of the remote screen the gaze is centred in remote window, but the gaze is not centred when looking at arbitrary direction. Therefore, an attempt was made for estimating gaze which seems to avoid this scenario in some cases, but it will require future work for further improvements.

VII. Future Work

As a future work, synthesizing gaze for online communication when someone is looking at arbitrary direction can help in building a robust gaze synthesis and correction system. The model can be further enhanced to add attention layers or even transformers, [18] to capture the eye gaze angles precisely. Secondly, unsupervised generative adversarial networks can be explored further because unsupervised learning does not require the annotated data with specific gaze angle and head pose information, and amount of data required can be reduced to a large extent [8], [12]. However, when training a model with GAN, [19] we might need to overcome issues related to artefact generation, as data distribution in realistic images are quite complex. Another practical change would be to build a gaze correction system which is platform agnostic so that the community can benefit to study this topic further irrespective of the platform they are working on. Lastly, it will be useful to study the effect of multiple participants at both local and remote side on gaze synthesis for online communication.

VIII. Conclusion

Gaze synthesis is pivotal for online communication, and this paper covers the study and comparison of different hyperparameters and deeper convolutions, along with the performance of models with subjective evaluation, and quantitative measurements. The experiments reveal quality gaze synthesis in real time for online communication is achievable when eye shape and colour is considered. However, there is still scope of improvement for complex eye structure.

Acknowledgment

I am thankful to Dr. Miles Hansard for the support and encouragement throughout this project. The constructive feedback were crucial in shaping the research and dissertation.

REFERENCES

[1] Awad D, Emery NJ and Mareschal I (2019) The Role of Emotional Expression and Eccentricity on Gaze Perception. Front. Psychol. 10:1129. doi: 10.3389/fpsyg.2019.01129.

[2] E. Wood, T. Baltrušaitis, L. P. Morency, P. Robinson, and A. Bulling. 2018. GazeDirector: Fully articulated eye gaze redirection in video.

[3] Y. Ganin, D. Kononenko, D. Sungatullina, and V. Lempitsky. 2016. DeepWarp: Photorealistic Image Resynthesis for Gaze Manipulation.

[4] Hsu, C.F., Wang, Y.S., Lei, C.L. and Chen, K.T., 2019. Look at me! Correcting eye gaze in live video communication. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM).

[5] Isikdogan, F., Gerasimow, T. and Michael, G., 2020. Eye Contact Correction using Deep Neural Networks. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.

[6] Greer, R. and Dubnov, S., 2021. Restoring Eye Contact to the Virtual Classroom with Machine Learning. arXiv preprint arXiv:2105.10047.

[7] Smith, B.A., Yin, Q., Feiner, S.K. and Nayar, S.K., 2013, October. Gaze locking: passive eye contact detection for human-object interaction. In Proceedings of the 26th annual ACM symposium on User interface software and technology (pp. 271-280).

[8] Zhang, J., Sun, M., Chen, J., Tang, H., Yan, Y., Qin, X. and Sebe, N., 2019. Gazecorrection: Self-guided eye manipulation in the wild using self-supervised generative adversarial networks. arXiv preprint arXiv:1906.00805.

[9] Kononenko, D. and Lempitsky, V., 2015. Learning to look up: Realtime monocular gaze correction using machine learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4667-4675).

[10] Jones, E.A. and Carr, E.G., 2004. Joint attention in children with autism: Theory and intervention. Focus on autism and other developmental disabilities, 19(1), pp.13-26.

[11] Kononenko, D. and Lempitsky, V., 2015. Learning to look up: Realtime monocular gaze correction using machine learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4667-4675).

[12] He, Z., Spurr, A., Zhang, X. and Hilliges, O., 2019. Photo-realistic monocular gaze redirection using generative adversarial networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6932-6941).

[13] Park, S., Mello, S.D., Molchanov, P., Iqbal, U., Hilliges, O. and Kautz, J., 2019. Few-shot adaptive gaze estimation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 9368-9377).

[14] Fischer, T., Chang, H.J. and Demiris, Y., 2018. Rt-gene: Real-time eye gaze estimation in natural environments. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 334-352).

[15] Kellnhofer, P., Recasens, A., Stent, S., Matusik, W. and Torralba, A., 2019. Gaze360: Physically unconstrained gaze estimation in the wild. In Proceedings of the IEEE/CVF International Conference on Computer Vision.

[16] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition

[17] He, K., Zhang, X., Ren, S. and Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision.

[18] Cheng, Y. and Lu, F., 2021. Gaze Estimation using Transformer.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in Proc. Advances Neural Information Processing Systems Conf., 2014.

[20] Mohammadi, M.R. and Raie, A., 2012, May. Robust pose-invariant eye gaze estimation using geometrical features of iris and pupil images. In 20th Iranian Conference on Electrical Engineering (ICEE2012).

[21] Cheng, Y. and Lu, F., 2021. Gaze Estimation using Transformer. arXiv preprint arXiv:2105.14424.

[22] Viola, P. and Jones, M., 2001, December. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition.

APPENDIX A
EVIDENCE

TABLE III
PSNR OF MODELS WITH DIFFERENT LOSS FUNCTIONS

DIRL		**CAVE**	
Loss	**PSNR**	**Loss**	**PSNR**
L2C	23.125	L2C	22.926
L2EB	23.112	L2EB	21.423
L2EL	23.125	L2EL	22.922
L2P	23.141	L2P	21.999
L2S	20.75	L2S	22.060
L2	23.113	L2	21.746
L2SC	23.151	L2SC	23.115

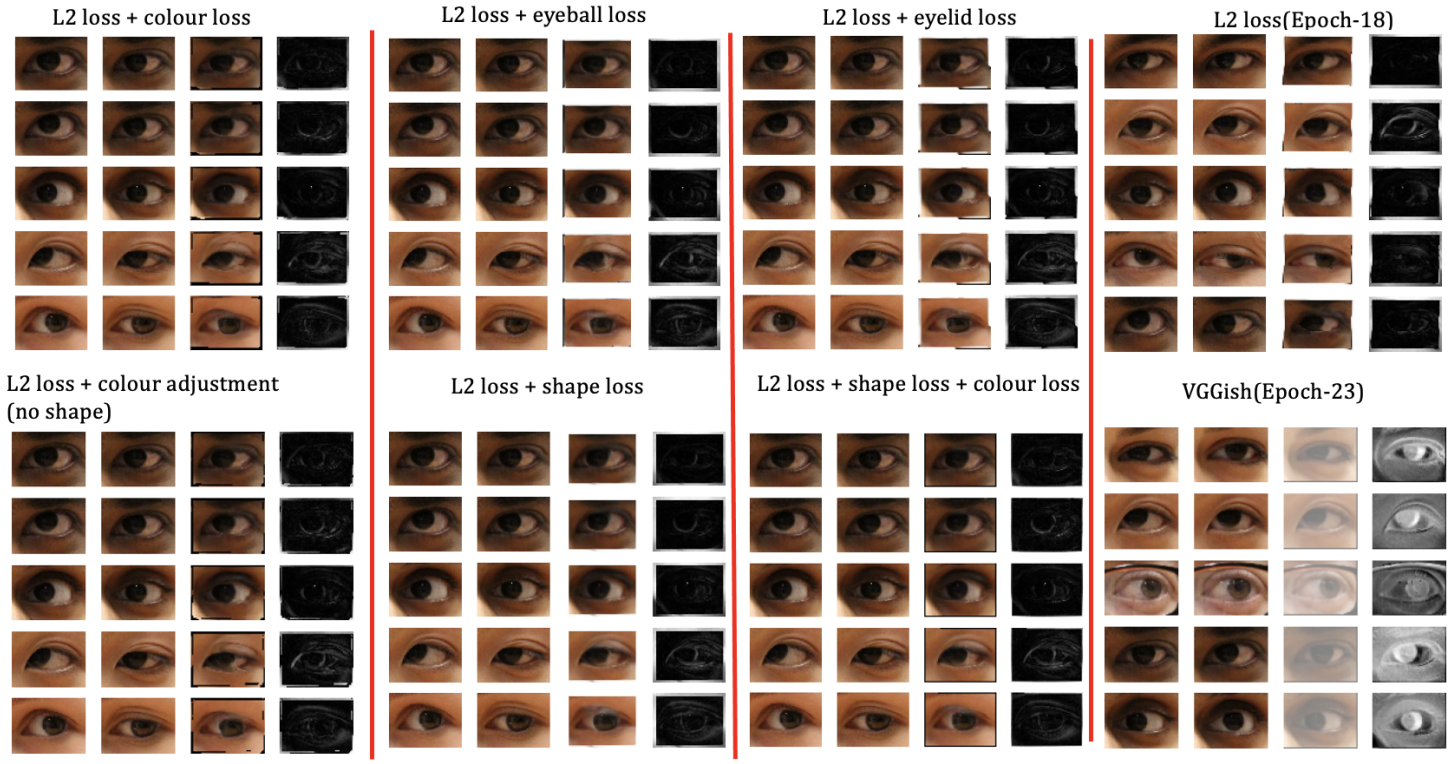


Fig. 14. CAVE Loss analysis, left to right in each subplot: input image, ground truth, synthesized image, difference between ground truth and synthesized image, lightness adjustment required

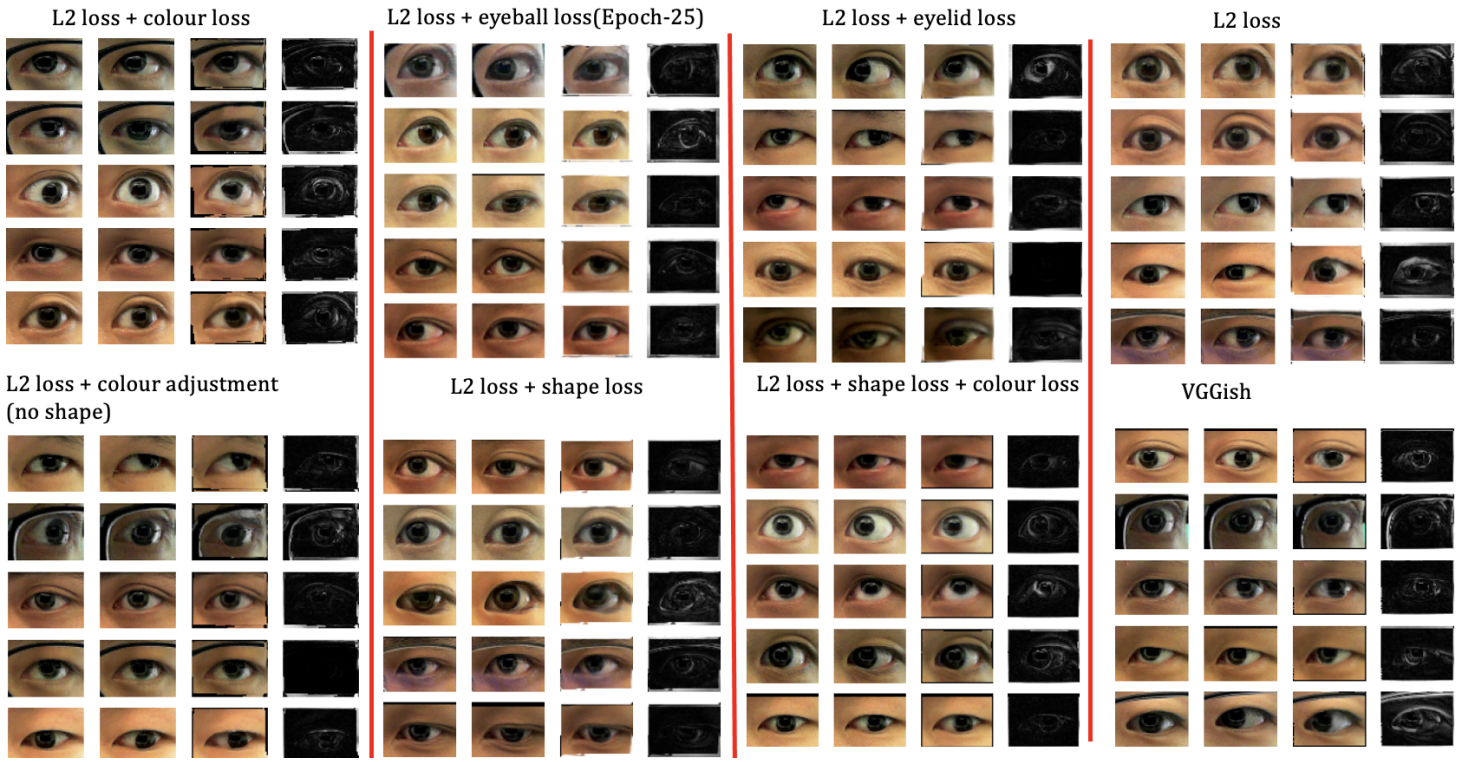


Fig. 15. DIRL Loss analysis, left to right in each subplot: input image, ground truth, synthesized image, difference between ground truth and synthesized image, lightness adjustment required

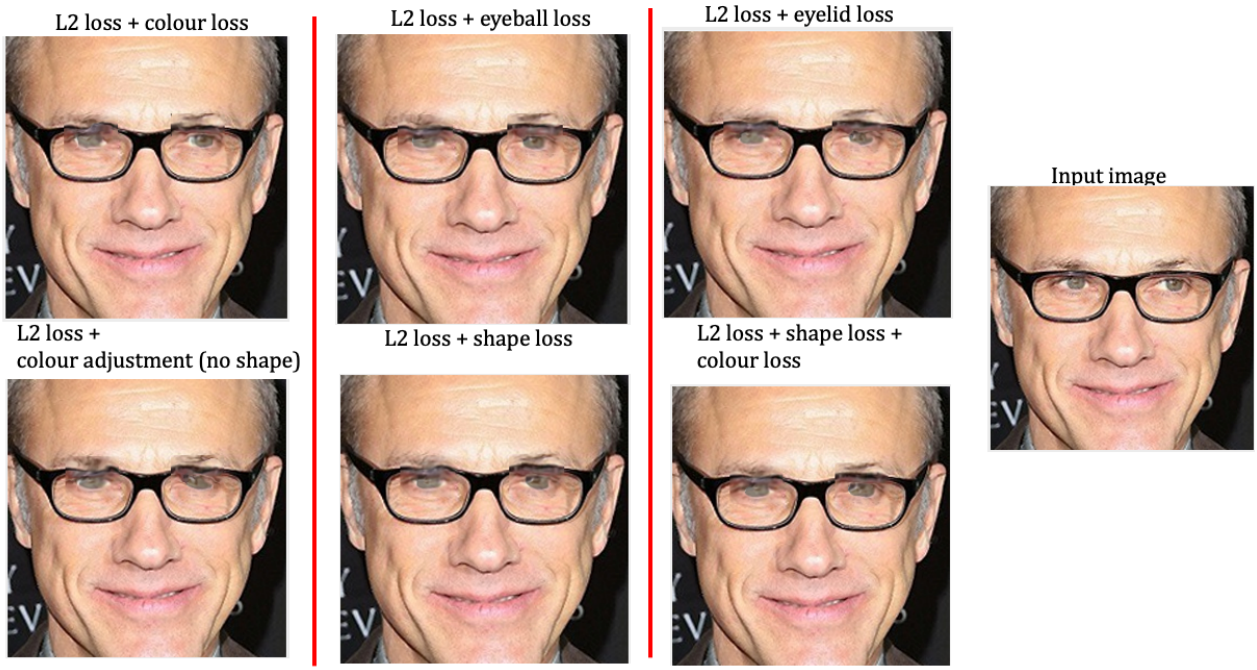


Fig. 16. Straight look + Spectacles

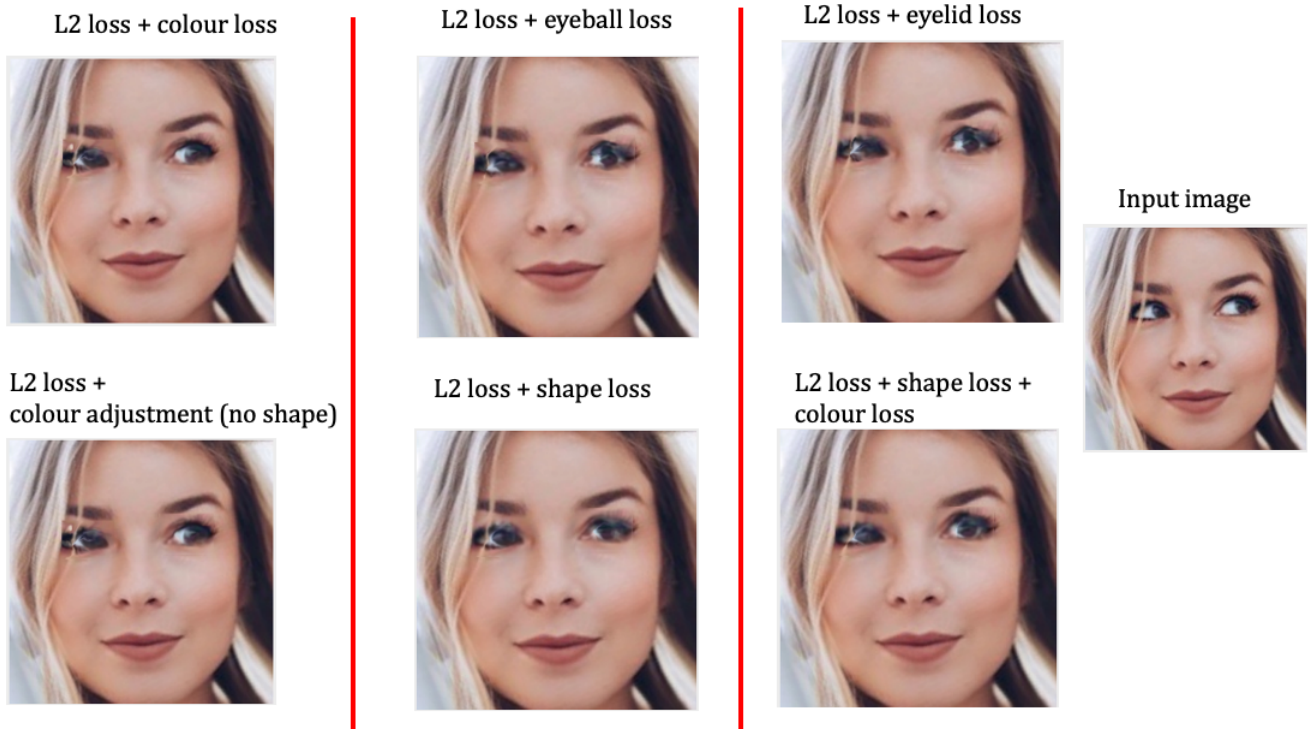


Fig. 17. Corner look + occluded with hair

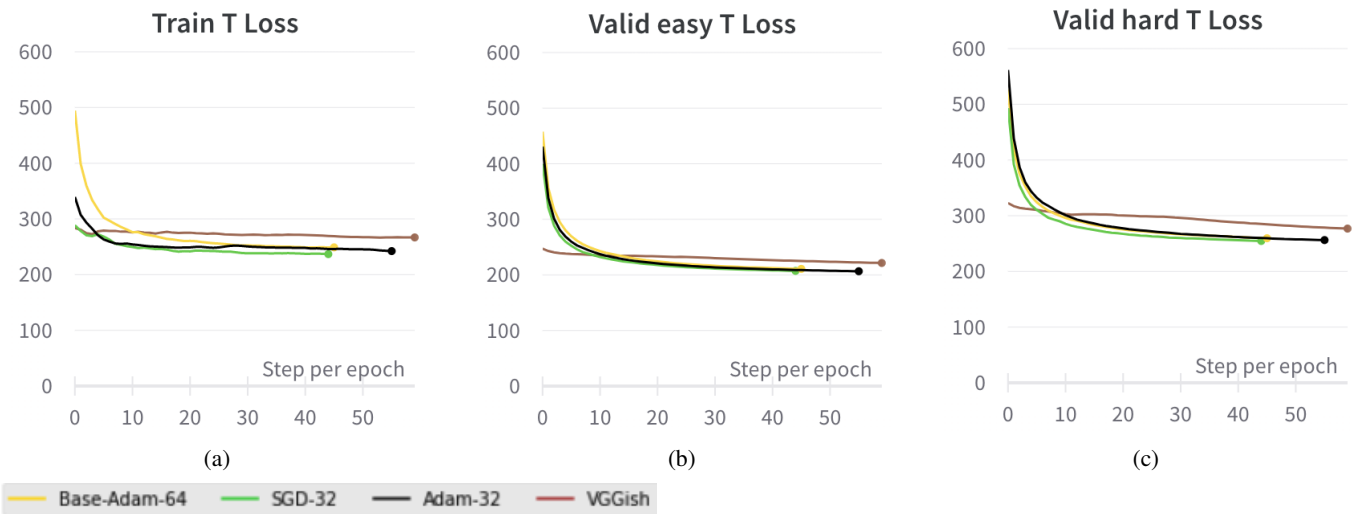


Fig. 18. Comparison of training and validation total losses for CAVE dataset: the total loss with SGD optimizer is the lowest. VGGish network with Adam optimizer continues learning. The base model with batch size of 64 continues learning, converges early as compared to batch size of 32.