Image Transfer Between Magnetic Resonance Images and Speech Diagrams

Kang Wang

A thesis submitted in partial fulfillment of the requirements for the
Degree of Master of Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

Realtime Magnetic Resonance Imaging (MRI) is a method used for human anatomical study. MRIs give exceptionally detailed information about soft-tissue structures, such as tongues, that other current imaging techniques cannot achieve. However, the process requires special equipment and is expensive. Hence, it is not quite suitable for all patients.

Speech diagrams show the side view positions of organs like the tongue, throat, and lip of a speaking or singing person. The process of making a speech diagram is like the semantic segmentation of an MRI, which focuses on the selected edge structure. Speech diagrams are easy to understand with a clear speech diagram of the tongue and inside mouth structure. However, it often requires manual annotation on the MRI machine by an expert in the field.

By using machine learning methods, we achieved transferring images between MRI and speech diagrams in two directions. We first matched videos of speech diagram and tongue MRIs. Then we used various image processing methods and data augmentation methods to make the paired images easy to train. We built our network model inspired by different cross-domain image transfer methods and applied reference-based super-resolution methods—to generate high-resolution images. Thus, we can do the transferring work through our network instead of manually. Also, generated speech diagram can work as an intermediary part to be transferred to other medical images like computerized tomography (CT), since it is simpler in structure compared to an MRI.

We conducted experiments using both the data from our database and other MRI video sources. We use multiple methods to do the evaluation and comparisons with several related methods show the superiority of our approach.

Table of contents

List of Figures

List of Tables

List of Acronyms

MRI	Magnetic Resonance Imaging
GAN	Generative Adversarial Networks
SRNTT	Super-Resolution by Neural Texture Transfer
WGAN	Wasserstein Generative Adversarial Network
RELU	Rectified Linear Unit
ELU	Exponential Linear Units
SGD	Stochastic Gradient Descent
AdaGrad	Adaptive Gradient
SISR	Single Image Super-Resolution
Ref SR	Reference-based Super-resolution
CNN	Convolutional Neural Network
SRCNN	Super-resolution Convolutional Neural Network
SRNTT	Super-resolution by Neural Texture Transfer
IQA	Image Quality Assessment
LCC	Linear Correlation Coefficient
SROCC	Spearman's Rank Order Correlation Coefficient
PLCC	Pearson Linear Correlation Coefficient
IS	Inception Score
DKL	Kullback-Leibler Divergence
MSE	Mean Squared Error

PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity
MSE	Mean Squared Error
MAE	Mean Absolute Error

1 Introduction

Nowadays, an increasing number of people are willing to learn at least one foreign language for a fellowship, travel, oversea study, or oversea work. Machine learning is an up and coming technology which brings benefit to different industries. We want to use machine learning to analyze tongue structure to contribute to linguistic study.

Magnetic Resonance Imaging (MRI) is a medical method with significant advantages in linguist study. First, it can reflect soft tissues like tongues and lips quite clearly, while CT scans work better on hard tissue such as bones. Second, MRIs scan in real-time, which means researchers can easily study continuous images. Third, when using MRI scan, researchers can choose their view from any direction. Apart from these benefits, MRIs are a relatively safe method since it is radiation-free.

Speech diagram is like the outline of speech related internal structure of a person's head. In a speech diagram, the jaw, tongue, lips, larynx, soft palate, and uvula are clearly shown. Generally, it looks like the semantic segmentation of an MRI. Our goal is to generate speech diagrams from MRIs and backwards using machine learning methods. The traditional way to build speech diagrams from MRI images requires plenty of manual work. The way to make it can be found in the *Seeing speech* website [1].

1.1 Image Transfer from Speech Diagrams to MRIs

For the translation from speech diagrams to MRIs, we firstly manually get our data from *Seeing speech* website [1] which has matched videos of tongue speech diagram and tongue MRI. Then we build our custom Generative Adversarial Networks (GAN) to train the data and do cross-domain image transfer between speech diagram and MRI. By using our custom GAN, we fulfill the translation from tongue speech diagram to tongue MRI.

1.2 Image Transfer from MRIs to Speech Diagrams

For the translation from MRIs to speech diagrams, we use the same dataset and custom GAN in 1.1. By cross-domain image transfer and other methods like data augmentation, we can train data of speech diagrams and paired MRI images and generate new speech diagrams from MRI videos of different pronouncers. Beginners of a new language can make progress through learning the skills shown on generated speech diagrams. To make the training process more efficient and the generation more accurate, we also use some image enhancing methods like image processing method black/white threshold and data augmentation. In the end, we obtain a positive result that satisfies our goal.

1.3 Thesis Contribution

In our proposed methods, we have two technical contributions that can improve the performance and efficiency of the translation task between speech diagrams and MRIs.

First, we build our custom GAN structure, which is inspired by Disco GAN [2], Dual GAN [3], and other cross-domain image transformation methods. This network is built to better generate MRI images from speech diagrams. The basic idea of cross-domain image transformation is to do translation from two different kinds of images, for example from human faces to cartoon avatars. To solve the low-resolution issue for the result data, we use image Super-Resolution by image Super Resolution by Neural Texture Transfer (SRNTT) [4] as our super-resolution method. SRNTT extracts features of target images and reference images and then does the texture replacement from the reference images to generated ones to make them in higher resolution. In our case, the reference images are also the ground truth images, so SRNTT is a suitable method to be applied. Also, we use the loss function from Wasserstein Generative Adversarial Network (WGAN) [5] which is proven to be better than classic GAN loss function.

The second contribution is the data enhancing methods, which speed up the training process and improve our generated data. We use different methods, including

basic data augmentation methods, image processing, and other methods to make the training data suitable for our goal.

1.4 Thesis Structure

In the abstract and introduction, we briefly introduce our methods and targets. Chapter 2 is about the materials that correspond to our research, including the tongue and language study, MRI study, machine learning methods, and image processing methods in addition to our evaluation methods. In Chapter 3, our work such as data extracting, model building, and data augmentation can be found. Detailed test information of comparison to other methods and ablation studies can be found in Chapter 4. The result is shown in experimental and results portion. At the end of this paper, we have a conclusion that indicates the summary of our work.

2 Literature Review

2.1 The Tongue and Pronunciation

As the principal organ of pronunciation movement, the tongue plays a vital role in simulated pronunciation teaching, virtual hosting, and visualization of pronunciation tongue position. Therefore, it is particularly important to establish a realistic tongue model and provide practical learning feedback.

Tongue is an important organ for language study. Language includes phonetic system, vocabulary system and grammar system. Language is a unique communication tool of humans, which develops with the development of human society. Real-time visual feedback can help learners to complete the language learning and correction process effectively. Visualization of the target joint position or posture helps learners use feedback directly to adjust the pronunciation and make it correct [6].

The tongue muscle is rhabdomyolysis muscle, which can be divided into two kinds: the inherent tongue muscle and the extra lingual tongue muscle. The intrinsic tongue muscle refers to the muscles that make up the tongue itself, starting and ending in the tongue. The muscle fibers of the intrinsic tongue are divided into three types: longitudinal, transverse and vertical. When contracted, the tongue can be shortened, narrowed or thinned, respectively. The external muscle starts from the outside of the

tongue and stops at the inside. According to its anatomical function, the muscles called genioglossus, styloglossus and hyoglossus are relatively important [7].

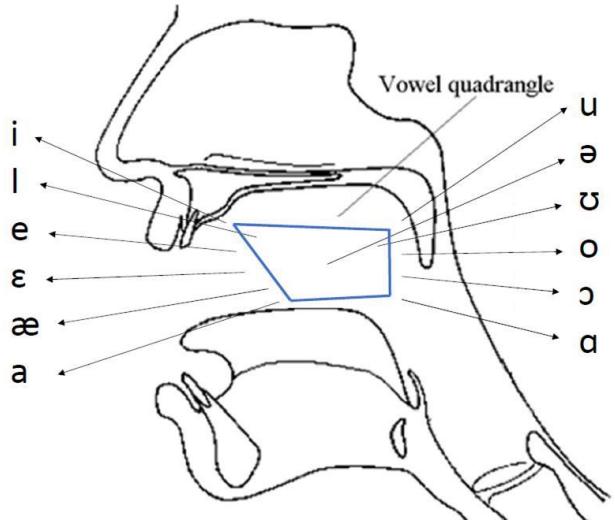

Figure 1Approximate tongue positions for vowel [8].

Figure 1 shows vowel quadrangle that illustrates approximate positions of the region tongue reaches when people speak vowels. When studying positions of tongue when pronunciation, watching it from side is easy for language learners to make associations and thus benefit from it.

The muscles of the tongue are flexible, and many common exercises are performed by one or more muscles together.

2.2 Magnetic Resonance Image

Magnetic Resonance Imaging (MRI) is a tomography technology which uses magnetic resonance to obtain electromagnetic signals from the human body and reconstruct the information of the human body. At first, it was widely used in physics, chemical biology and other fields, and it was not used in medical clinical testing until 1973[9]. MRI has also made great progress in the study of language pronunciation patterns and correctness by observing tongue movements [10].

In all medical imaging methods, MRI has high resolution of soft tissue contrast, which can clearly distinguish soft tissues such as muscle, tendon, fascia and fat. In addition, MRI has the ability of direct slicing in any direction. Combining with different directions of slicing, the structure of organs or tissues under examination can be fully displayed without blind angle of observation, and the body layer images of cross section, sagittal plane, coronal plane and various inclined planes can be directly made for convenient observation and research.

In the early stage, MRI is an important imaging diagnostic method for mental retardation. Conventional MRI can provide detailed anatomic and morphological information and can assist clinical diagnosis of etiology. For children with obvious nervous system diseases, MRI has important diagnostic value. Brain dysplasia and

delayed myelination of white matter are common MRI findings in children with language retardation [11].

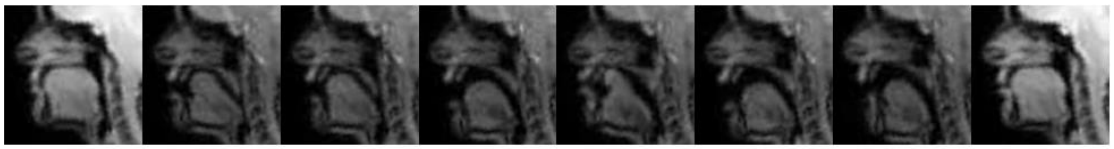

Figure 2 An example of real-time MRI

In aspect of language learning, or pronunciation, there are few researches directly related to tongue structure. Some Researches for relevant topic mainly focus on brain or connections between language and diseases. One research is about language and verbal memory after right hemispheric stroke [12].

Compared to CT, magnetic Resonance Imaging is generally considered as a better choice even X-ray damage of CT is now well controlled in most medical settings. For example, one chest CT scan delivers the number of X-rays in 100 to 200. MRI can produce different information than CT and is also widely used in hospitals and clinics. MRI scans may present risks and discomforts. MRI scans typically take longer than CT scans, and they typically require the subject to enter a narrow restriction tube. In addition, people with some medical implants or other non-removable metals in the body may not be able to safely undergo MRI. Besides, MRI was called nucleus magnetic Resonance Imaging, but to avoid negative associations the "nucleus" was removed [13].

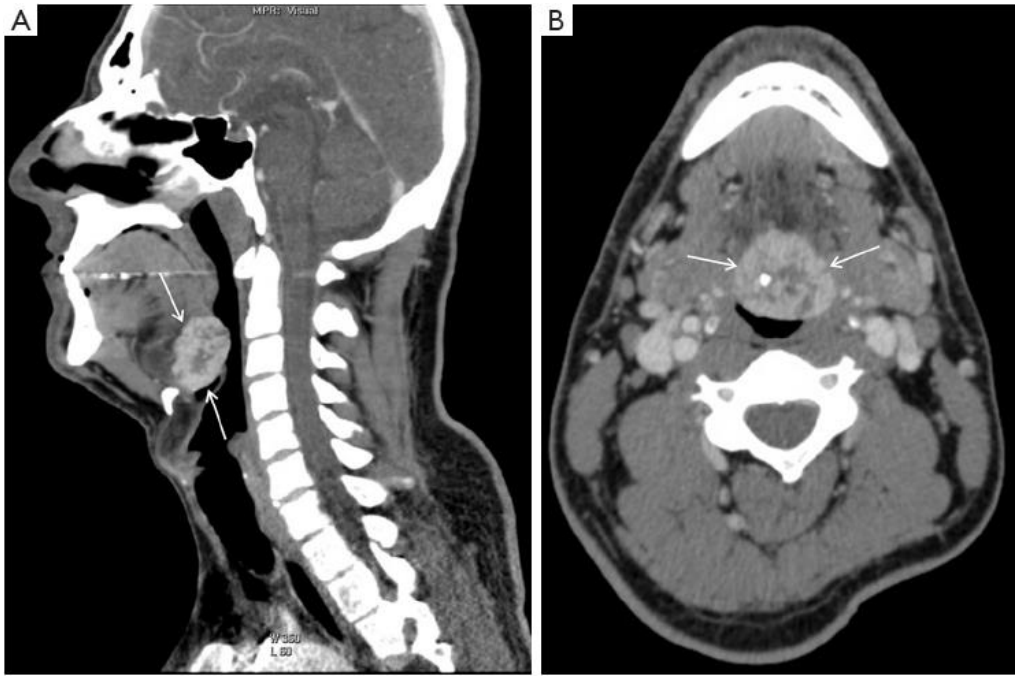

Figure 3 Top view and side view of brain CT scan [14]

MRI also has an advantage over ultrasound imaging. Ultrasound imaging was widely developed in the 1970s and has been widely used in ultrasound diagnosis, ultrasound therapy and biomedical ultrasound engineering. In addition to traditional and extensive medical uses, researchers use it for self-behavior recognition, learning and correction. By showing the contours of the tongue on a screen, the experiment allows viewers to get visual feedback on their own pronunciation, improving their pronunciation accuracy by comparing it to native speakers' behavior [15]. Based on the visual learning process, our method adopts a new neural network training system based on the original experiment to obtain more accurate and clear results. However, when taking ultrasound image of tongues, the result image only shows the frontal side. Besides, only tongues are represented so the connection between tongue and teeth and

other parts are barely known. In all, ultrasound image is not useful for language learners to gain experience from.

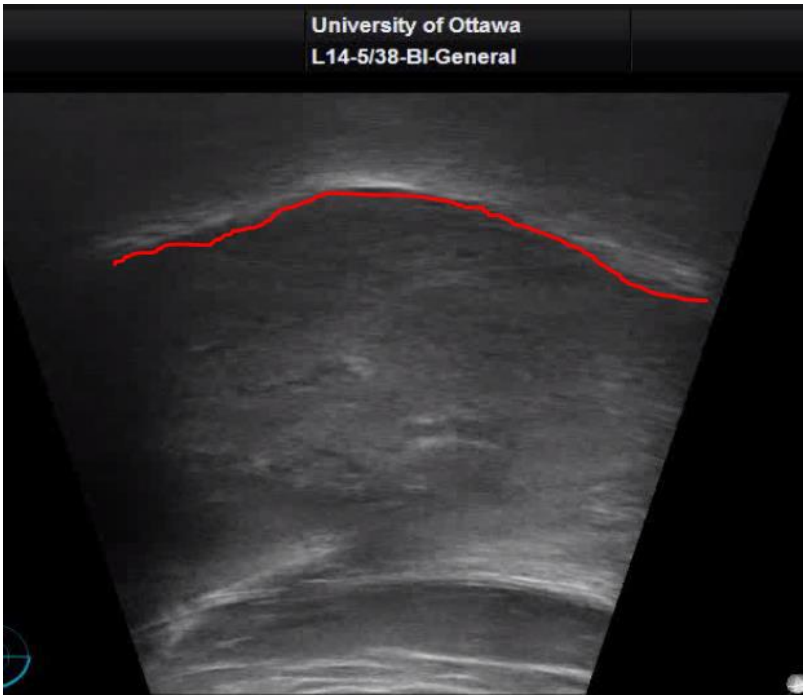

Figure 4 Ultrasound tongue image when pronouncing [15]. The red line shows the position of tongue

Besides, real time is the irreplaceable advantage of MRI. Our training and testing data are clips from videos and we need continued pictures to study the speaking or singing process of tongue. Because of this, real time is necessary to ensure the whole process of the tongue movement can be observed.

In recent years, many scholars have done research on pronunciation mode and pronunciation position. Research on visual information and language perception showed that /l/ and /n/ had similar pronunciation positions [16], but there were visual differences. Finally, this visual information helped people to improve the correct perception rate of English consonants [17].

2.3 Convolutional Neural Networks

2.3.1 Neuron and Neural Network

Neural networks are computer programs inspired by biological neural network. Neural networks can learn to approximate complex function mappings with properly designed architectures and enough training data.

The basic element of neural network is neuron. Neurons are functions which receive one or more inputs and produce an output. In general cases, each input is assigned with a weight. We add the sum of all input weights and a bias which is optional to the activation function to generate the output. This process is non-linear. Here is a general formulation to show this process:

$$y = f(\sum_{i=1}^{k} w_i x_i + b) \qquad 2.1$$

In this formulation y means the output, w_i refers to each weight of the input x_i. Bias is called as b and is not always used.

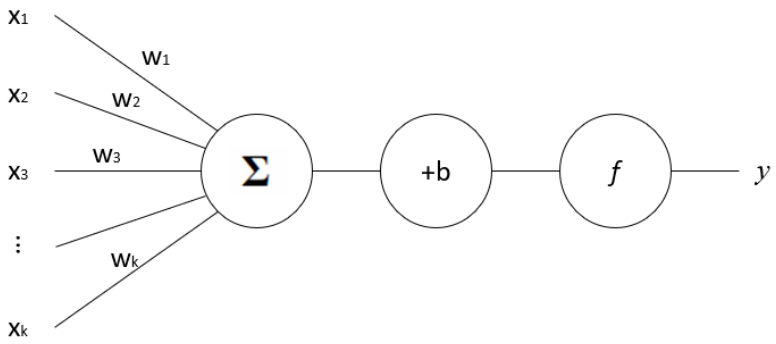

Figure 5 A typical neural in neural network

Then neural units are used in a chain structure which is named neural network. The first layer of neural network is called input layer and receive all the inputs. The computational data is passed from the previous layer to the next layer until to the last one which we call as output layer. Finally, the outputs are generated from output layer and the whole process is finished. Layers between the input layer and output layer are named hidden layers. The depth of the neural network is the total number of the layers.

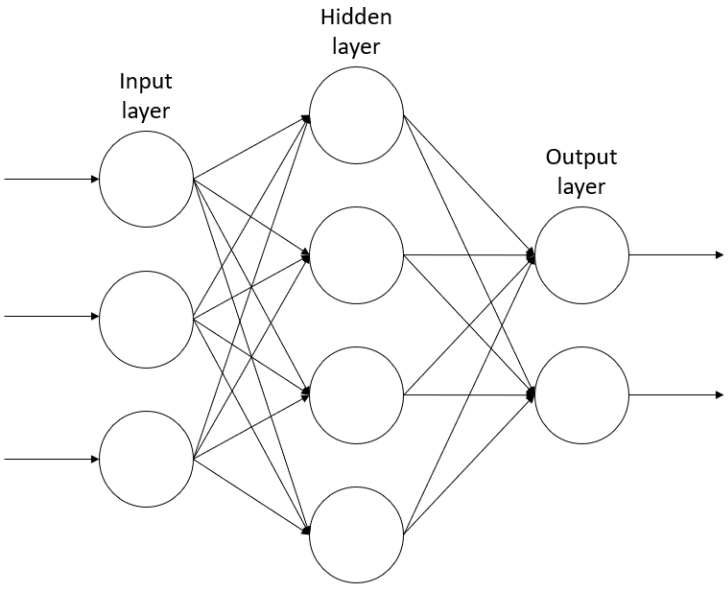

Figure 6 An example of neural network

2.3.2 Layers

2.3.2.1 Convolutional Layer

As we described in 2.3.1, in a neural network, a layer is formulated as:

$$y = f(\mathrm{W} \cdot \mathrm{X} + b) \tag{2.2}$$

where y is output, W is weights, X is input, and b is bias. Here the symbol · means matrix multiplication. In convolutional neural network, a layer is defined as:

$$y = f(\mathrm{W} * \mathrm{X} + b) \tag{2.3}$$

where symbol * means convolution operation.

Convolutional neural networks (CNN) was invented for computer vision tasks, and its input is assumed to be images. Regarding the weights being convolved into images, they can also be considered as filter coefficients. Therefore, the term weight in the context of CNN is usually called kernel. Processing the image through CNN is equivalent to extracting features from the input image by filters.

Convolutional layers are usually used to extract features from multi-channel images. The channels of input image and output image are called input channels and output channels. The size of the filter coefficients is called kernel size. Kernel size is often set to 3 × 3, 5 × 5 or 7 × 7.

The input of convolutional layer is often not zero-padded before it is fed. If not, the size of outputs will shrink. Zero padding input enables kernel size and the output size to be controlled independently, so there is no need to maintain balance between kernel size and network depth.

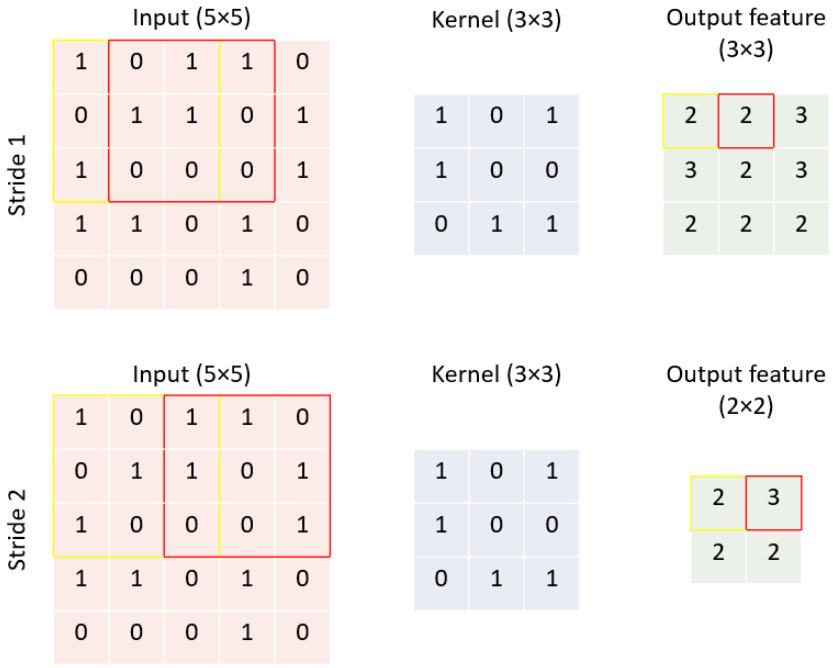

Figure 7 Convolution with stride 1 and 2.

The convolution operation slides the kernel over the entire image. The step of sliding in convolutional layer is called stride. The default convolution stride is 1, which means that the kernel slides one pixel at a time. Another common stride is 2, it is used to reduce the output size. Stride larger than 2 is uncommon used in practice.

2.3.2.2 Activation Layer

Usually, a convolutional layer is followed by an activation layer which aims to introduce nonlinearity to the network. In each activation layer we use none-linear

functions to deal with the sum of the weights. These functions are called activation functions. Plenty of activation functions are used in today's machine learning study. In our paper, we mainly introduce the most popular ones.

The first one is rectified linear unit (ReLU). The general ReLU function will only keep the positive part of the outputs and generate 0 is the outputs are below 0. This activation function aims to dismiss the effect of the negative data.

$$ReLU(x) = max\ (0, x) \tag{2.4}$$

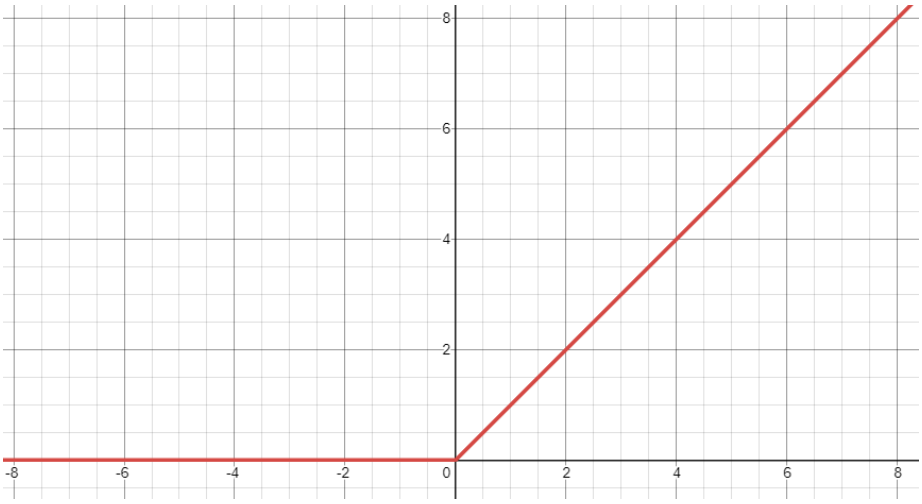

Figure 8 ReLU function

To reduce the effect of the negative data, other activation functions like leaky ReLU function and exponential linear units (ELU) function [18] use different methods. In Figure 8 we can see a small and non-zero gradient when x < 0 for leaky ReLU function. Leaky ReLU function is formulated as below:

$$Leaky\ ReLU(x) = \begin{cases} x & if\ x > 0 \\ 0.01x & otherwise \end{cases} \tag{2.5}$$

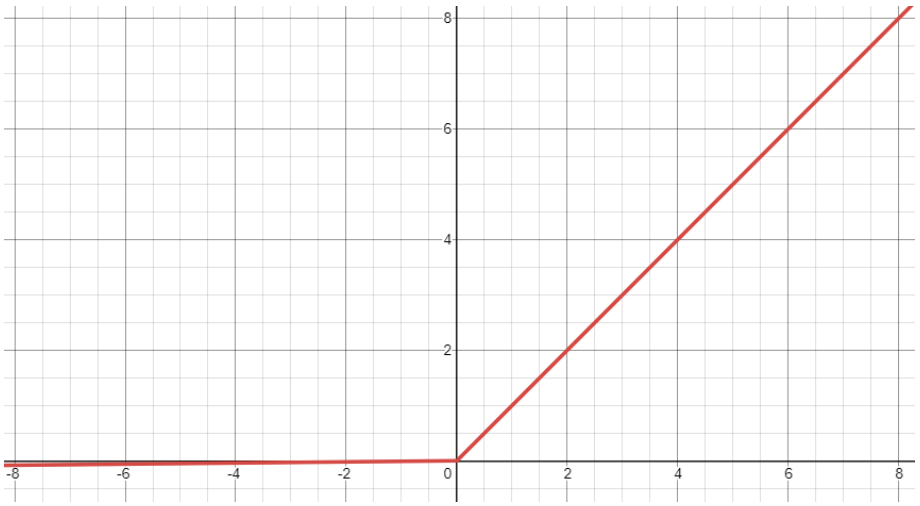

Figure 9 Leaky ReLU function

Exponential linear units (ELU) function [18] is more accurate than standard ReLU in classification. It is defined as:

$$ELU(x) == \begin{cases} x & if\ x > 0 \\ a\,(e^x - 1) & otherwise \end{cases} \qquad 2.6$$

where a is a positive hyper-parameter to be tuned.

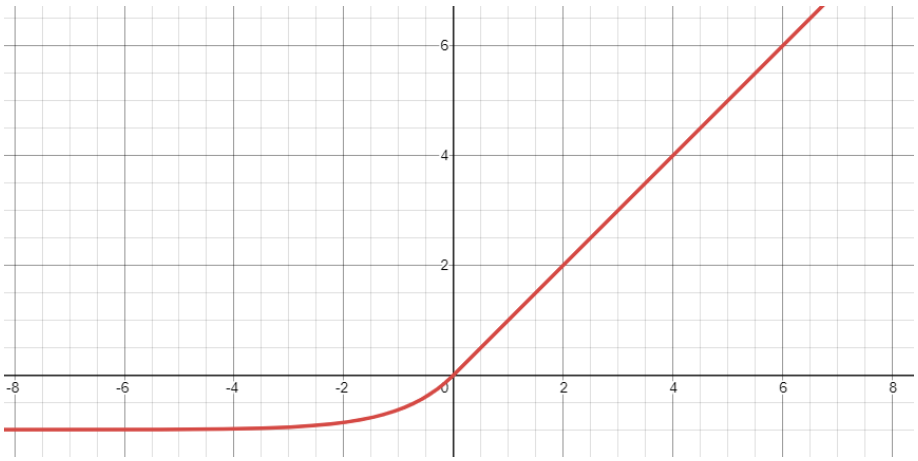

Figure 10 ELU function

Sigmoid function is also commonly used. Its curve is like character “s” and its output range is within [0,1]. Sigmoid activation function is often used in a binary classier when the target class is often expressed with 0 or 1. It is formulated as:

$$\text{Sigmoid}(x) = \frac{1}{1+e^{-x}} \quad 2.7$$

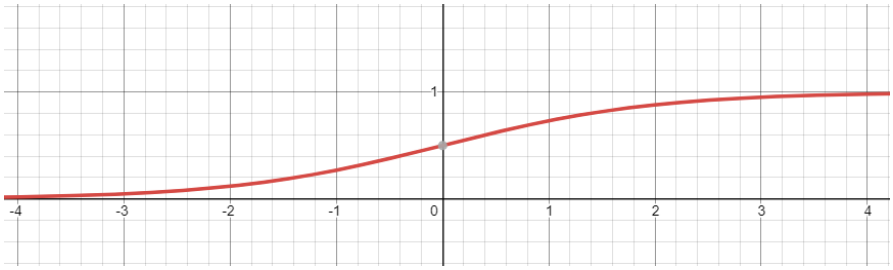

Figure 11 Sigmoid function

2.3.2.3 Pooling Layer

The pooling operation is like the convolution operation. Pooling is applied one patch at a time, and then slide the pooling kernel along the width and height directions of the input feature map. There are two types of pooling layers: max pooling layer and average pooling layer. The max pooling layer remains the max value of each patch of the feature map and discard the rest of the information in the patch. The average pooling layer generates the average value in each patch.

The most common set of pooling layers is 2 × 2 for kernel size and 2 for slide. This means each time the pooling operation will get one value from 4 values. The output size of pooling layer can be formulated as:

$$0 = (N - F)/S + 1 \quad 2.8$$

where O is output size, N is input size, F is pooling kernel size and S is stride.

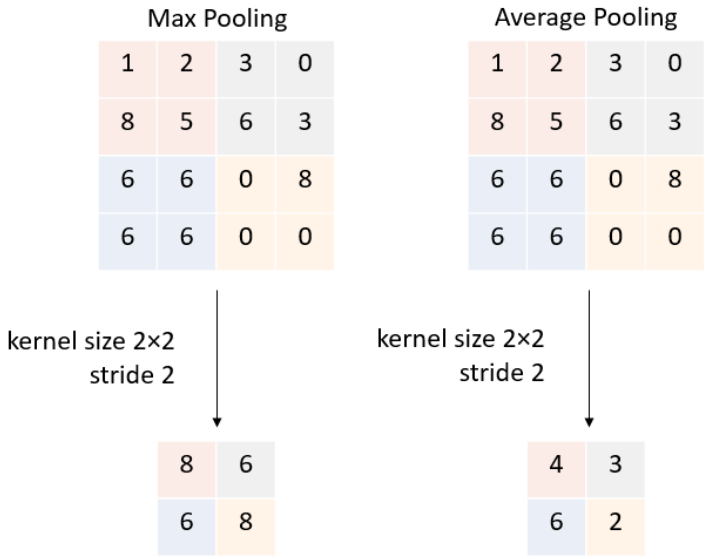

Figure 12 Max pooling vs. average pooling.

Recent works [19] show that pooling layers can be replaced by convolutional layers with increased stride. The replacement can be beneficial for the computation of networks because the learnable kernels in convolutional layers is more flexible than regular pooling layers.

2.3.3 Training Methods

2.3.3.1 Dataset

The data to support machine learning workflow is usually split as three datasets: training data, test data and validation data.

Firstly, the training data is shown to the model. The learning algorithm adjusts the model parameters to fit the training data. Then, the fitted model makes predictions on the validation data set. The validation data set gives an unbiased estimate of the training progress. Finally, the test data set is used to measure the performance of the trained model.

2.3.3.2 Parameter and hyper-parameter

In a machine learning model, there are parameters and hyper-parameters which control behaviors of the model.

Parameters are tunable. The process of training the model is to find the best parameters with the help of the training data. In a convolutional neural network, each layer has two kinds of parameters: weights and biases. The total number of parameters is the sum of all weights and biases.

Hyper-parameters are not adapted by the learning process. They include variables such as number of hidden units to determine the network structure, and the variables such as learning rate to determine how the network is trained.

2.3.3.3 Loss Function

A loss function is a function that maps a network evaluation onto a real number intuitively representing loss value associated with the evaluation. The network optimization seeks to minimize a loss function [20]. In this section we will introduce some commonly used loss functions.

L1 loss is the sum of absolute errors. It is formulated as:

$$\mathrm{L1}(\mathrm{I},\mathrm{I}_{\mathrm{gt}}) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I_{\mathrm{gt}}(i,j) - I(i,j)| \qquad 2.9$$

where I_{gt} is the ground truth image and I is the predicted image. M * N is their image size. $|\cdot|$ means the absolute value.

Similarly, Mean Absolute Error (MAE) is defined as:

$$\mathrm{MAE}(\mathrm{I},\mathrm{I}_{\mathrm{gt}}) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I_{\mathrm{gt}}(i,j) - I(i,j)| \qquad 2.10$$

L2 loss is the sum of squared errors whose formulation is:

$$\mathrm{L2}(\mathrm{I},\mathrm{I}_{\mathrm{gt}}) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I_{\mathrm{gt}}(i,j) - I(i,j)]^2 \qquad 2.11$$

MSE is short for the mean squared error, which is defined as:

$$\mathrm{MSE}(\mathrm{I},\mathrm{I}_{\mathrm{gt}}) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I_{\mathrm{gt}}(i,j) - I(i,j)]^2 \qquad 2.12$$

The cross-entropy loss function is often used to quantify differences between two probability distributions. It is a popular loss function for classification, but not suitable for image processing problems. The cross-entropy loss is formulated as:

$$\text{Cross Entropy (p, q)} = -y \log\hat{y} - (1 - y) \log (1 - \hat{y}) \qquad 2.13$$

where p and q are the true and predictive probability distributions and p $\in$ {y, 1-y}, q $\in$ {$\hat{y}$, 1-$\hat{y}$}.

2.3.3.4 Forward Propagation

For forward propagation, the input data passes through the input layer, hidden layer and output layer and the network output is directly obtained. The final output can be represented as:

$$F(x) = f_n(f_{n-1} \dots f_i\ (\dots f_1\ (x))) \tag{2.14}$$

where x is the input of networks and the f_i is function of each network layer.

2.3.3.5 Back propagation

Back propagation [21] refers to an algorithm that allows information to flow backward through the network from the loss value in order to calculate the gradient more efficiently [22] than forward propagation.

The back-propagation algorithm is based on the chain rule of calculus. Chain rule is used to calculate the derivatives of variables in the nested equations. As an example, let a nested function be F(x) = $f_2(f_1(x))$. The chain rule indicated that:

$$\frac{dF}{dx} = \frac{df_2}{df_1} \cdot \frac{df_1}{dx} \tag{2.15}$$

In convolutional neural network, the derivative of loss of weights is calculated according to the chain rule. Let E be the loss, s_j^k, a_j^k be the output of convolutional

layer and activation layer of node *j* in layer *k*, and w_{ij}^{k} be the weight between node *i* in layer *k-1* and node *j* in layer *k*. The partial derivative of loss of w_{ij}^{k} is:

$$\frac{\partial \mathrm{E}}{\partial w_{ij}^{k}} = \frac{\partial \mathrm{E}}{\partial a_{j}^{k}} \cdot \frac{\partial a_{j}^{k}}{\partial s_{j}^{k}} \cdot \frac{\partial s_{j}^{k}}{\partial w_{ij}^{k}} \qquad 2.16$$

The partial derivative of loss of weights can be calculated by chain rule.

2.3.3.6 Optimization

Optimizing the neural network means minimizing the loss function L(x). Therefore, the optimization of the network is to find a set of weights to ensure that the loss of the network reaches the lowest value. The optimization algorithm in CNN helps to reduce loss effectively.

Gradient descent is a basic method of optimization algorithms in neural networks. The formula for gradient descent is:

$$\Theta = \Theta - \eta \cdot \nabla J(\Theta) \qquad 2.17$$

where Θ is the parameters, η is the learning rate and $\nabla J(\Theta)$ is the gradient of loss $J(\Theta)$.

Stochastic gradient descent (SGD) is an extension for gradient decent algorithm. It is faster than gradient descent, and it is formulated as:

$$\Theta = \Theta - \eta \cdot \nabla J(\Theta; x_i, y_i) \qquad 2.18$$

where x_i and y_i are the training input the corresponding desired output at i-th iteration. Although updated frequently can help find better local minima, it also leads to fluctuation.

SGD with momentum [23] utilizes the previous update to calculate the current one. It can be defined as:

$$\triangle \Theta_t = \alpha \cdot \triangle \Theta_{t-1} - \eta \cdot \nabla J(\Theta) \quad 2.19$$

Where α is the momentum, η is the learning rate and $\nabla J(\Theta)$ is the gradient of loss $J(\Theta)$.

The problem of SGD with momentum is that at the end of training process, the network almost gets the minima while the momentum is still too high. Compared with classical SGD, SGD with momentum prevents fluctuations by trying to stay on the same direction.

Adaptive gradient (AdaGrad) [24] introduces per-parameter learning rate to traditional SGD. Compared to traditional SGD, AdaGrad uses different learning rate at for different parameters. The formula of parameter updating is:

$$\Theta_{i,t+1} = \Theta_{i,t} - \frac{\eta}{\sqrt{G_{ii,t}+\varepsilon}} \cdot g_{i,t} \quad 2.20$$

where t is time step, η is the initial learning rate, $g_{i,t}$ is the gradient of $\Theta_{i,t}$, ε is a small number to avoid divided-by-zero issue. $G_{ii,t}$ is formulated as:

$$G_{ii,t} = \sum_{\tau=1}^{t} g_{i,\tau}{}^2 \quad 2.21$$

where g is the gradient at iteration τ.

2.4 Autoencoder and Variational Autoencoder

2.4.1 Autoencoder

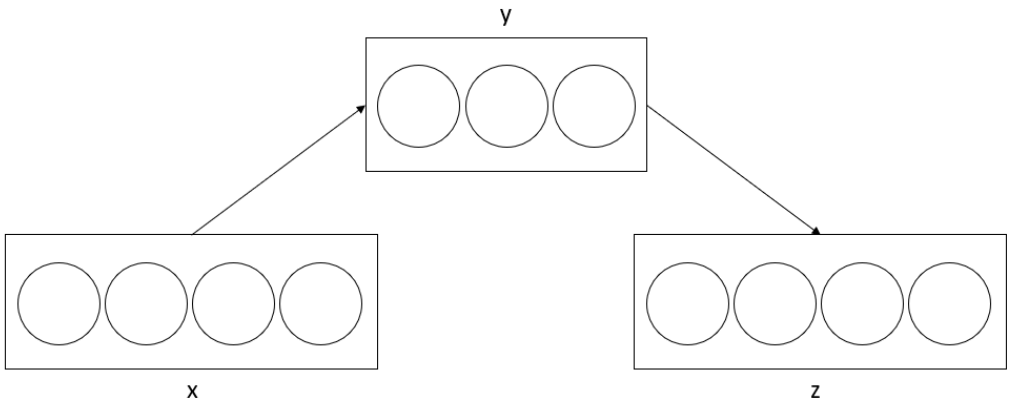

Figure 13 Traditional autoencoder [25]

The traditional autoencoder [26] contains encoder and decoder. Both of encoder and decoder refer to multi-layer networks. It is an artificial neural network that attempts to reproduce its input. The output data and the input data are of the same type. The basic idea of autoencoder is mapping essential structure in the input $x_{(i)}$ to $y_{(i)}$. The compressed representation has lower dimensionality than the input. In the case of image recognition $x_{(i)}$ might be an image (pixels) while $y_{(i)}$ might consist of edge in various orientations.

However, for traditional autoencoder the latent space's regularity is not ensured. Some points of the latent space may give meaningless content when decoded, thus leading to an overfitting. If one does not care about definition of the architecture, the

network may work on any overfitting possibilities so we cannot rely on it to generate the data we want.

2.4.2 Variational Autoencoder

The two most used methods to generate images are variational autoencoder [27] and Generative Adversarial Network. Variational autoencoders is an autoencoder with regularized training encodings distribution that has good properties in latent space to generate data. The variational inference method and the regularization of it has close relationship. It is an autoencoder whose training is regularized to avoid overfitting and ensure that the latent space has good properties that enable generative process. The regularity means continuity and completeness. In variational autoencoders after decoding, two close points in the latent space should give contents that are not too different, and all points sample from latent space should give meaningful contents. The process of optimizing properties of distribution can be formulated as:

$$P_\theta(x^{(i)})=\int P_\theta(x^{(i)}|z)P_\theta(z)dz \quad 2.22$$

$$\Theta^*=\arg\max_\theta \sum_{i=1}^{n} log\ P_\theta(x^{(i)}) \quad 2.23$$

where P_θ is the distribution where input is mapped in, θ is its properties, the retionship between input x and the latent space encoding vertor z is revealed by pior $P_\theta(z)$ and likelihood $P_\theta(x|z)$. Θ* is the optimal properties.

Compared with autoencoders, variational autoencoders have the same architecture as traditional autoencoders but will output two vectors from the encoder,

one can be treated as a mean vector, and anther is a variance vector. Meanwhile, error sampled from Gaussian distribution is also used together with mean and variance to be composited as code. Noise is used when computing the code, it is multiplied by the variance and then added to mean. Because noise is randomly generated, the progressive graph will be generated where the noise intervals coincide so the images generated by variational autoencoder can be better in continuity and completeness.

2.5 Generative Adversarial Network (GAN)

2.5.1 Typical GAN

The typical Generative Adversarial Network (GAN) [28] has two networks, one is the generator and the other one is the discriminator, which are inspired by the two-person zero-sum game and can achieve the best generation effect by two networks confronting each other.

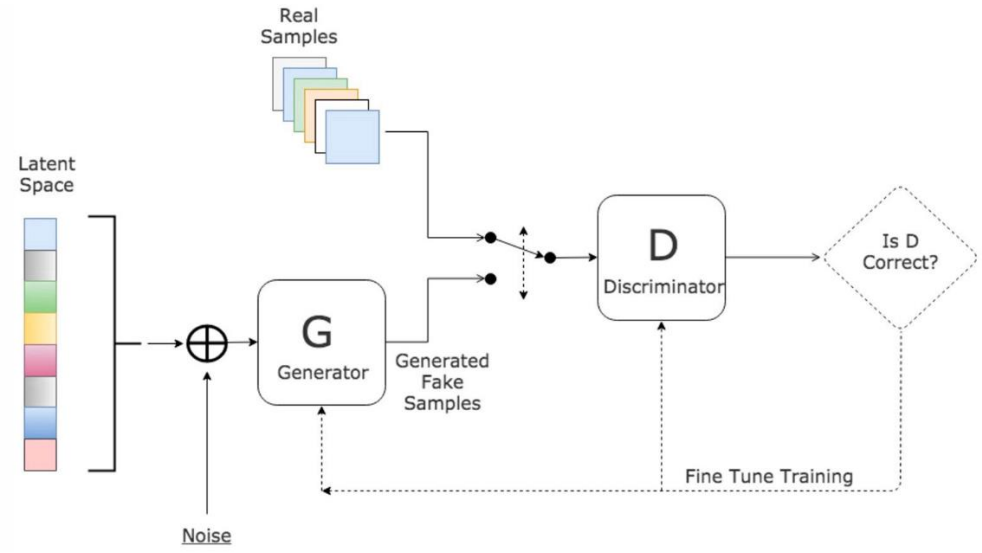

Figure 14 The architecture of typical GAN [28]

The generator G is to generate fake data to fit the potential distribution of real data, while the discriminator D is to distinguish real data from fake data according to the distribution difference of fake data. The input of the generator is a random noise vector. The noise is mapped to a new data space by the generator to obtain a multi-dimensional vector which is called fake data. The discriminator D is a binary classifier that takes both the real data and the fake data generated by the generator as input. The output of the discriminator represents the probability that the sample is a real sample.

The typical GAN is unsupervised. The min-max optimization of a typical gan is defined as:

$$\min_G \max_D E_{x\sim P_r(x)}[\log(D(x))] + E_{x\sim P_g(x)}[\log(1-D(\tilde{x}))] \quad 2.24$$

where x is from the real data distribution $P_r(x)$ and $\tilde{x}$ is from the prior distribution $P_g(x)$. $D(\tilde{x})$ can also be written as D(G(z)) where z is the random noise.

2.5.2 Conditional GAN

Conditional GAN [29] is an important example to illustrate the variety of Generative Adversarial Nets. The original GAN is unsupervised and may generate random results. In Conditional GAN, they construct the network by simply feeding a data y, which means the condition and is used both in generator and discriminator. By this method the data are labeled and thus the network can provide designated outputs.

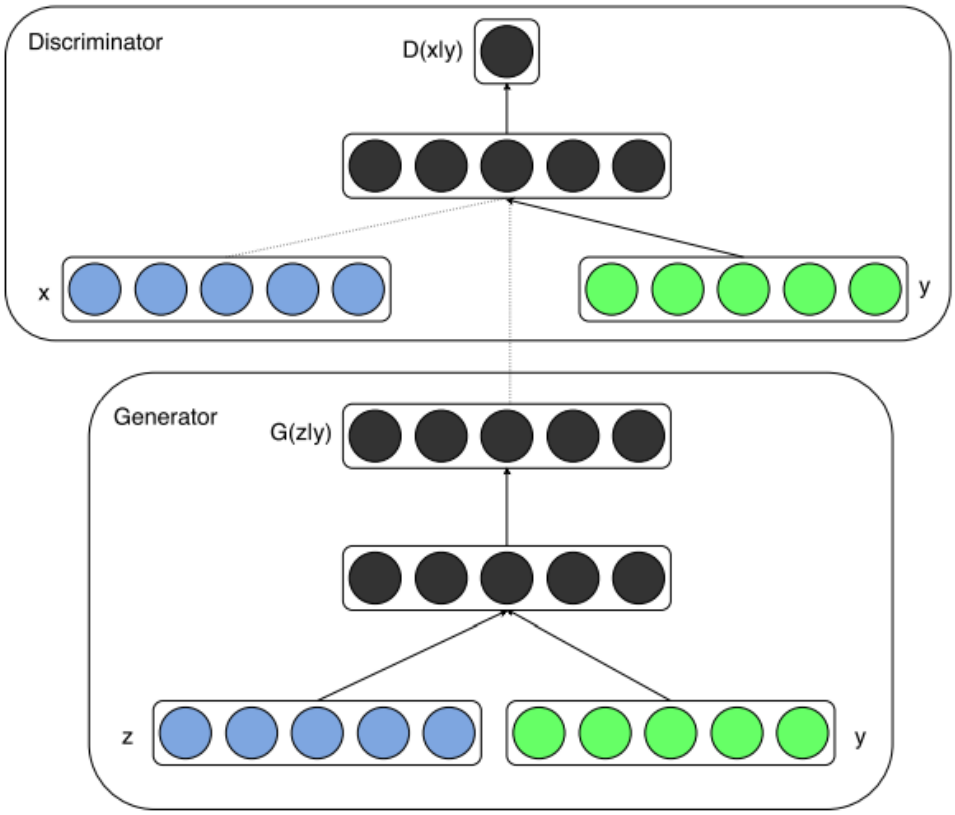

Figure 15 Conditional adversarial net [29]

The objective function of Conditional GAN min-max optimization is formulated as:

$$\min_G \max_D E_{x\sim P_r(x)}[\log(D(x|y))] + E_{z\sim P_z(z)}[\log(1-D(G(z|y)))] \quad 2.25$$

where x is sampled from real data set $P_r(x)$, y is the condition variable, z is from the prior input noise $P_z(z)$.

2.5.3 GAN Applications in Cross-Domain Image Transfer

Cross-domain image transfer means to transfer images of different kinds. For example, from hand bag images to shoe images. In this section we will introduce some typical GAN Applications in cross-domain image transfer.

Disco GAN works well in cross-domain image transfer. Using the discovered relations, Disco GAN can successfully transfer style from one domain to another and meanwhile preserves attributes like orientation and face identity.

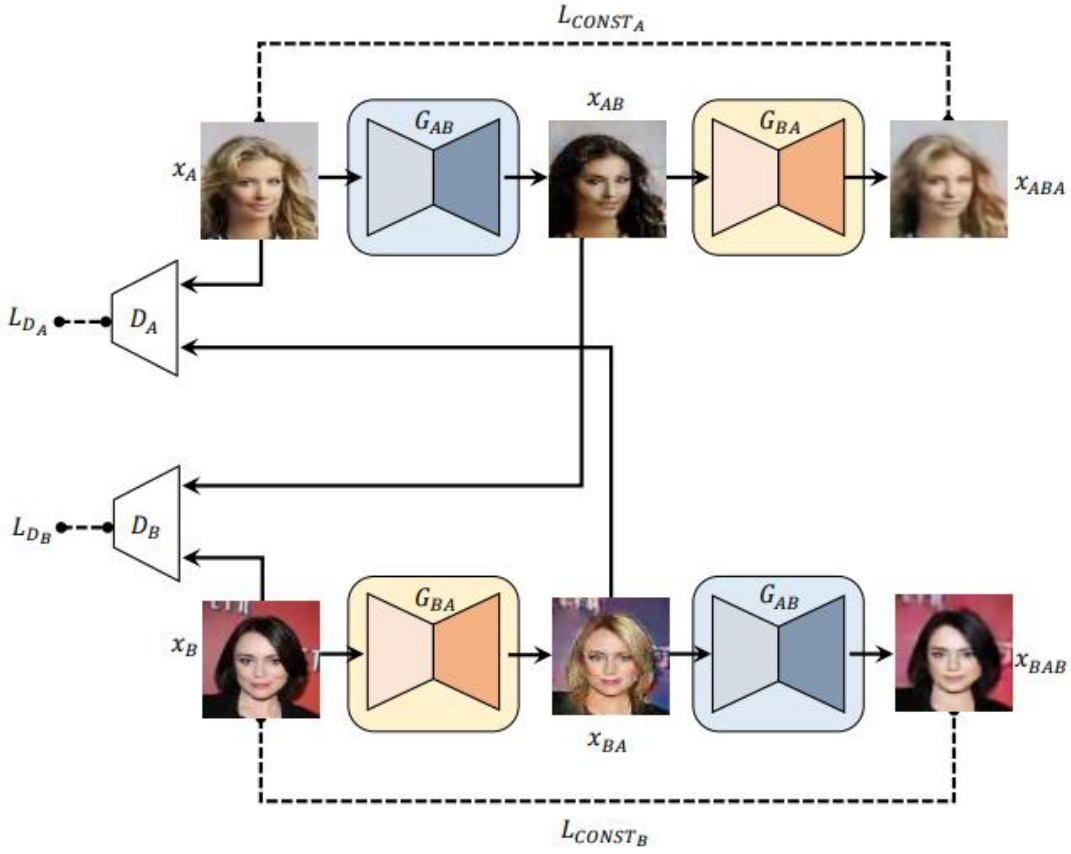

Figure 16 The architecture of Disco GAN [2]

Unlike typical GAN, Disco GAN has two generators G_{AB} and G_{BA}, and associated adversarial discriminators D_B and D_A. The input image X_A in domain A is translated to X_B from domain B through the generator G_{AB}. The second generator G_{BA}

translates X_B into a domain A image X_{ABA} which matches the original input image X_A. This process can be defined as:

$$X_{AB} = G_{AB}(X_A) \quad 2.26$$

$$X_{ABA} = G_{BA}(X_{AB}) = G_{BA}(G_{AB}(X_A)) \quad 2.27$$

L_{CONST_A} is the reconstruction loss that compares the input image X_A and the reconstructed image X_{ABA}. It can use any form of metric function like L1 and L2 mentioned in 2.3.3.3. It can be formulated as:

$$L_{CONST_A} = d(G_{BA}(G_{AB}(X_A)), X_A) \quad 2.28$$

where d means the distance between the input image and the reconstructed image.

The generators and discriminators of Disco GAN use the standard generator loss and the standard discriminator loss. The total generator and discriminator loss of it can be formulated as:

$$L_G = L_{G_{AB}} + L_{G_{BA}}$$

$$= L_{GAN_B} + L_{CONST_A} + L_{GAN_A} + L_{CONST_B} \quad 2.29$$

$$L_D = L_{D_{AB}} + L_{D_{BA}} \quad 2.30$$

Dual GAN [3] is another widely used method in cross-domain image transfer. Dual GAN and Disco GAN both use an unsupervised learning method for cross-domain image transfer, but their loss functions are different[30].

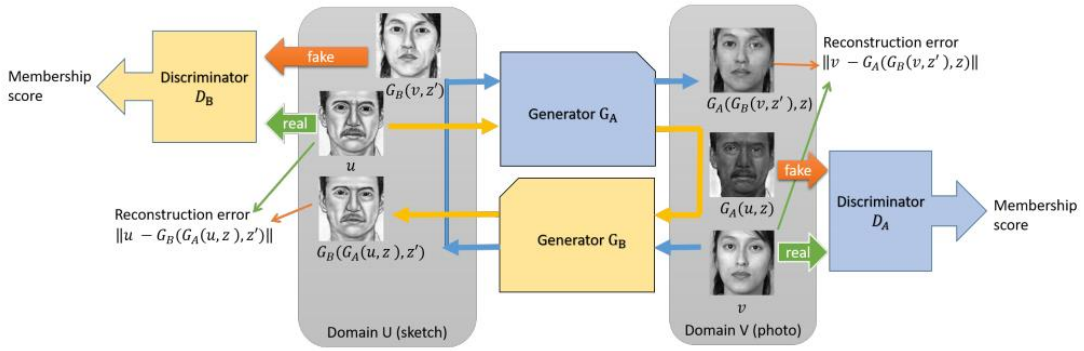

Figure 17 Network architecture and data flow chart of DualGAN

Dual GAN has a similar principle as Disco GAN. The data flow and the total loss of it can also be formulated as Eq 2.26, Eq 2.27, Eq 2.29, Eq 2.30. While its generator loss L_G and discriminator loss L_D are defined as:

$$L_G = \lambda_U \|u - G_B(G_A(u,z),z')\| + \lambda_V \|v - G_A(G_B(v,z'),z)\| - D_B(G_B(v,z')) - D_A(G_A(u,z)) \quad 2.31$$

$$L_D = D_A(G_A(u,z)) - D_A(v) + D_B(G_B(v,z')) - D_B(u) \quad 2.32$$

where u and v are input images from domain V and domain U, λ_U and λ_V are two constant parameters. λ_U and λ_V are set to [100.0, 1,000.0] according to the input image type of the application.

Cycle GAN is widely used in style transfer such as collection style transfer, object transfiguration, season transfer, and photo enhancement. It can be thought of as a paint-style transfer technique that captures the specific features of one data set and figures out how those features translate to other image data sets in the absence of paired training sets. This problem can be broadly referred to as image migration, where images can be transferred from one scene to another.

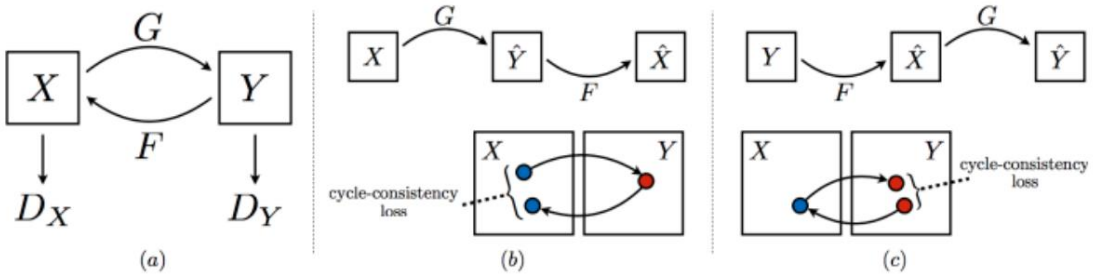

Figure 18 Illustration for the CycleGAN [31] model

Cycle GAN adopts GAN with cycle consistency loss for cross-doamin image transfer task. The cycle consistency loss of Cycle GAN is similar to the reconstruction loss of Disco GAN. For the mapping function G, the objective is expressed as:

$$L_{GAN}(\mathrm{G.\ D_Y,\ X,\ Y}) = E_{y\sim p_{data}(y)}[\log D_{\mathrm{Y}}(\mathrm{y})] + E_{x\sim p_{data}(x)}[\log(1 - D_{\mathrm{Y}}(\mathrm{G(x)}))] \quad 2.33$$

where G aims to generate images G(x) that look like images from domain Y, D_Y is the discriminator that distinguishs between generated samples G(x) and real samples y.

The cycle consistency loss can be formulated as:

$$L_{cyc}(\mathrm{G.\ F}) = E_{x\sim p_{data}(x)}[\|\mathrm{F(G(x))\text{-}x}\|_1] + E_{y\sim p_{data}(y)}[\|\mathrm{G(F(y))\text{-}y}\|_1] \quad 2.34$$

The full objective is:

$$L\ (\mathrm{G.\ F,\ D_X,\ D_Y}) = L_{GAN}(\mathrm{G.\ D_Y,\ X,\ Y}) + L_{GAN}(\mathrm{F.\ D_X,\ Y,\ X}) + \lambda L_{cyc}(\mathrm{G.\ F}) \quad 2.35$$

where λ is used to set the importance of the two objectives.

2.6 Image Super Resolution

Since low resolution images cannot be beneficial enough for language learners, we tried some super resolution methods. For the traditional single image super-resolution (SISR) problem, the perceived constraints improve the image quality and make the image look clearer, but the fake textures and artifacts are produced.

Super-Resolution is to improve the resolution of the original image by hardware or software. The process of obtaining a high-resolution image through a series of low-resolution images is super-resolution reconstruction. The core idea of super-resolution reconstruction is to exchange time bandwidth, which is acquired from multi-frame image sequence of the same scene for spatial resolution, to realize the conversion from temporal resolution to spatial resolution. In the past three decades, most algorithms have gradually shifted from frequency-based to spatial-based. For example, based on feedforward depth network, each neuron in the network starts from the input layer, receives the previous input, and inputs to the next level, until the output layer.

For the traditional single image super-resolution (SISR) problem, the perceived constraints improve the image quality and make the image look clearer, but the hallucinate fake textures and artifacts are produced.

Super-resolution convolutional neural network (SRCNN) [32] is one of the first works which introduces Convolutional Neural Network (CNN) to SR. It has three stage frameworks patch extraction, non-linear mapping and reconstruction.

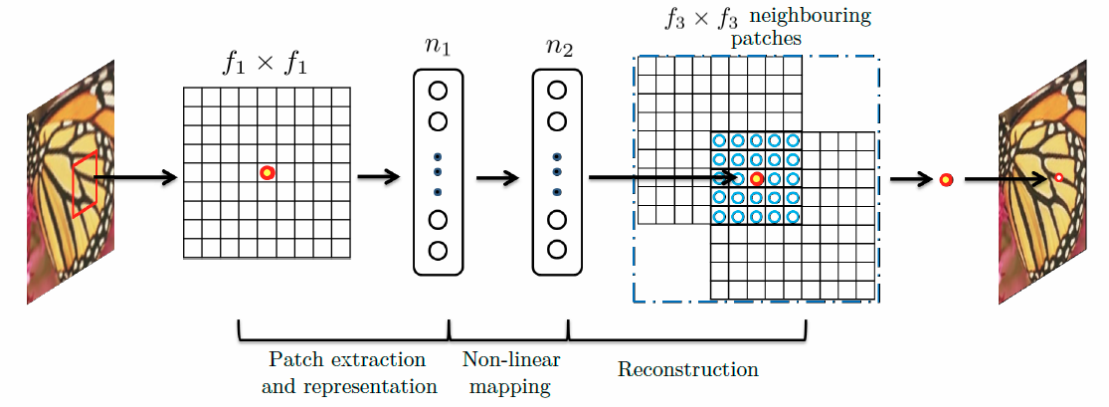

Figure 19 The architecture of the SRCNN [32]

First, they extract various features from low-resolution images in the form of high dimensional vectors. These vectors are called feature maps. Then they transform these feature maps to another set of feature maps non-linearly. At last, the feature maps produced by the non-linear mapping are aggregated to the high-resolution ground truths images.

Image super-resolution by neural texture transfer (SRNTT) [4] is also a reference-based approach, known as reference-based super-resolution (Ref SR), which compensates for lost detail in low-resolution images by taking advantage of rich textures in the high-resolution reference image. However, previous methods require reference images to contain similar content to low-resolution images and require image alignment, otherwise these methods will be ineffective. To address the shortcomings of

traditional reference-based super-resolution methods, SRNTT does not require image alignment, but rather transferring semantically relevant features by matching them in feature space.

In this method, reference data is also needed. While doing the transformation from low-resolution to high-resolution, a reference data means high-resolution images that is the same kind of input image. By using the texture from the reference data, the generated high-resolution images will look more natural and real.

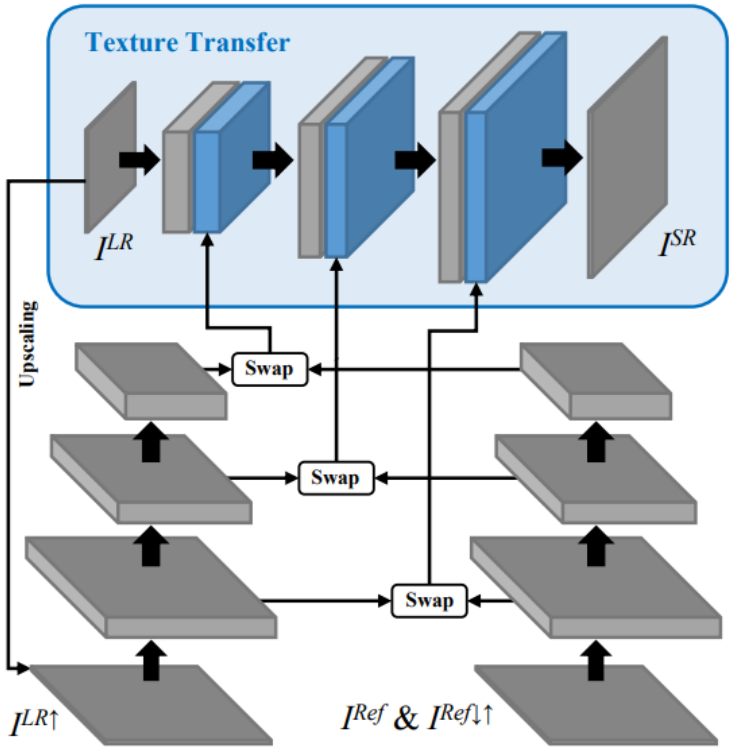

Figure 20 The proposed SRNTT [4] framework with feature swapping and texture transfer.

The proposed SRNTT aims to use the low-resolution input images I^{LR} and the reference images I^{Ref} to estimate the super- resolution images I^{SR}. The basic idea is to

extract matching texture from I^{Ref} and transfer it to I^{SR}. This process is called Texture Transfer. The upscaled low-resolution images I^{LR} is get by bicubic up-sampling method. Bicubic downsampling and up-sampling are applied to get the $I^{Ref\downarrow\uparrow}$ that matches the frequency band of $I^{LR\uparrow}$.

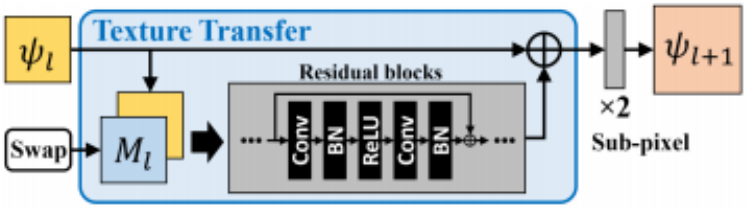

Figure 21 The network structure for texture transfer [4]

The residual blocks in the Texture Transfer part aims to extract related texture from M_l conditioned on ψl and merge it with target content. Sub-pixel convolution [37] is used in the end of this process to donate 2 times upscaling. The whole process can be defined as:

$$I^{SR} = Res(\psi_{L-1} || M_{L-1}) + \psi_{L-1} \quad 2.36$$

where ψ_{L-1} is the network output at layer l-1, M_{L-1} is the related reference images, Res(·) is the residual blocks and || is channel-wise concatenation.

In general cases, it will make the generation results more real but may not be accurate. For instance, if we use the low-resolution summer street scene picture as input but we use a high-resolution spring street scene picture as reference image. The generated high-resolution street scene picture may make people feel it is in spring.

2.7 Image Quality Evaluation

Image Quality Assessment (IQA) is generally divided into 3 categories according to the amount of information provided by the original reference image: Full Reference-IQA (FR-IQA), Semi-Reference (Reduced Reference-IQA, RR-IQA) and No Reference (No Reference-IQA, NR-IQA). FR-IQA has both original and distorted images and is less difficult to evaluate [33]. The core is to compare the information content or feature similarity of the two images.

There are many indicators for measuring the image quality evaluation results, and each indicator has its own characteristics. Usually, the differences and correlations between the objective values of the model and the subjective values of the observation are compared.

The two common evaluation indexes are Linear Correlation Coefficient (LCC) and Spearman's Rank Order Correlation Coefficient (SROCC). LCC, also known as Pearson Linear Correlation Coefficient (PLCC), describes the linear correlation between subjective and objective assessment, as defined below:

$$\mathrm{LCC} = \frac{\sum_{i=1}^{N}(y_i-\bar{y})(\hat{y}_i-\bar{\hat{y}})}{\sqrt{\sum_{i=1}^{N}(y_i-\bar{y})^2}\sqrt{\sum_{i=1}^{N}(\hat{y}_i-\bar{\hat{y}})^2}} \qquad 2.37$$

Where N represents the number of distorted images, y_i and $\hat{y}_i$ represent the true value and prediction of the i-th image respectively, and $\bar{y}$ and $\bar{\hat{y}}$ represent the true mean value and the predicted mean value respectively.

SROCC measures the monotonicity of the algorithm's prediction, and the calculation formula is:

$$SROCC = 1 - \frac{6\sum_{i=1}^{N}(vi-pi)^2}{N(N^2-1)} \quad 2.38$$

Where v_i and p_i represent the positions of y_i and $\hat{y}_i$ in the real values and predicted value sequence respectively.

The Inception Score (IS) [34] is a metric for automatically evaluating the quality of image generative models. This metric was shown to correlate well with human scoring of the realism of generated images from the CIFAR-10 dataset. The IS uses an Inception v3 Network pre-trained on ImageNet and calculates a statistic of the network's outputs when applied to generated images.

$$IS(G)=exp\ (E_{x\sim p_g} D_{KL}(p(y|x)||p(y))) \quad 2.39$$

$x \sim p_g$ indicates that x is an image sampled from p_g

DKL(p||q) is the Kullback-Leibler divergence between the distributions p and q

$$D_{KL}\ (P||Q) = \sum_i P(i) log \frac{P(i)}{Q(i)} \quad 2.40$$

p(y|x) is the conditional class distribution

$$p(y) = \int_x p(y|x)p_g(x) \quad 2.41$$

is the marginal class distribution.

There are many approaches proposed to compare the processed images with the ground truth. Three quality evaluation metrics we used for our project are Mean squared error (MSE), Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM).

The Peak Signal to Noise Ratio (PSNR) is usually used to evaluate the quality of an image compared with the original image after compression. The higher the PSNR, the smaller the distortion after compression. It can be calculated by means of Mean Square Error (MSE):

$$MSE\ (Is,\ Ig) = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[Ig(i,j) - Is(i,j)]^2 \qquad 2.42$$

$$PSNR\ (Is,\ Ig) = 10 \cdot log_{10}[\frac{(MAX-MIN)^2}{MSE\,(Is,Ig)}] \qquad 2.43$$

where M * N is image size, Ig is the ground truth image and Is is the predicted image.

PSNR is the most widely used performance quantization method in the field of image and video processing. The problem of PSNR is that it is affected by pixel points..

Due to the limitations of PSNR, the image quality reflected by PSNR is not completely consistent with the image quality observed by the Human eye in some situations, and some important physiological, psychological and physical characteristics are not considered.

The basic idea of SSIM is to evaluate the similarity of two images from the following three aspects: luminance, contrast and structure.

$$SSIM\ (x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma} \quad 2.44$$

Where x and y stand for images, α,β,γ are weights. The $l(x, y), c(x, y)\ and\ s(x, y)$ are the comparison measurements on luminance, contrast and structure respectively, and are defined as:

$$l(x, y) = (2\mu_x\mu_y+c_1)/(\mu_x^2+\mu_y^2+c_1) \quad 2.45$$

$$c(x, y) = (2\sigma_x\sigma_y+c1)/(\sigma_x^2+\sigma_y^2+c_1) \quad 2.46$$

$$s(x, y) = (\sigma_{xy}+c_3)/(\sigma_x\sigma_y+c_3) \quad 2.47$$

where μ_x and μ_y are local means, σ_x^2 and σ_y^2 are standard deviations, σ_{xy} is cross-covariance, and $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$, $c_3 = \frac{1}{2}c_2$. L is the dynamic range of the pixel-values, $k_1 = 0.01$ and $k_2 = 0.03$ by default.

SSIM improves on the shortcomings of PSNR. However, it cannot operate effectively when the image is shifted, scaled and rotated (all of which are unstructured distortions).

3 Methodology

3.1 Dataset

Our dataset has been obtained from the website *Seeing speech* [1]. For each position of vowel and consonant, we can find related speech diagram images and MRI images from videos shown on the website.

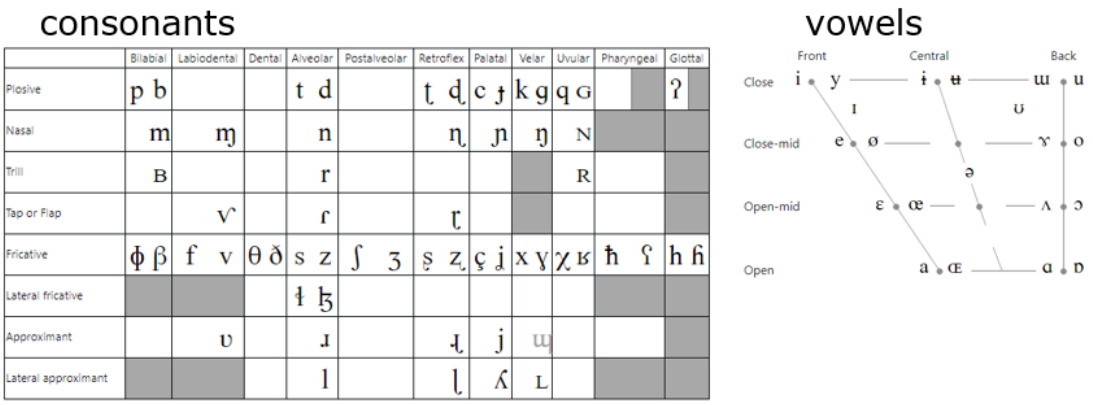

Figure 22 The dataset from Seeing speech (totally 21 vowels and 58 consonants) [1].

Speech diagram videos are created using parts of matched MRI videos. Hence the early seconds of the two different videos are totally matched. We have written a program to take snapshots of the matched videos and gone through the full dataset to create consistent and exact matching. If the program works too rapidly, many similar snapshots will be caught. On the other hand, if the time interval between two snapshots is lengthened, we lose many useful and informative data points. Ultimately, our program works every 0.5 seconds and we obtain a total of 1,906 paired data measurements.

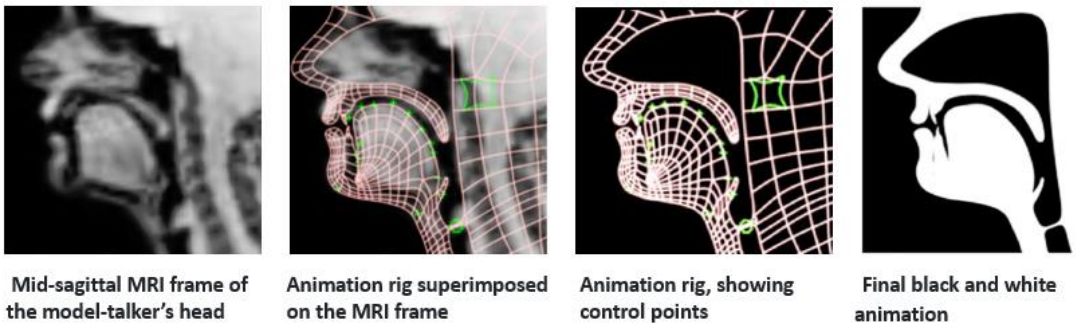

Figure 23 How speech diagram is created by MRI [1].

After the paired speech diagrams and MRI images were obtained, we then attempted different ways to make it easier for our model to train the data. In the original videos, the tongue in both speech diagram and in MRI are at different angles and in different shapes. The original video clips also have useless superfluous information which required removal. We studied the training data image location and relationship. To match the training dataset and the test dataset, rotation and cropping were utilized to be certain our model specifically focused on the head. Here we mainly focused on the head, making the head upright and of similar size.

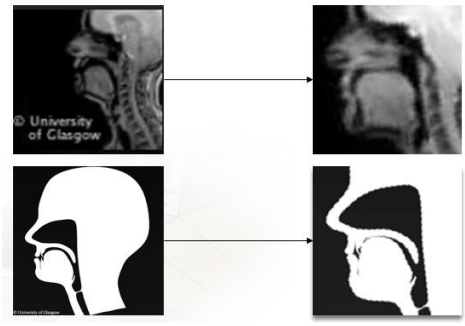

Figure 24 Images of MRI and speech diagrams are rotated and cropped to focus on the most salient feature

The training is too time consuming if we do the transformation from MRIs to speech diagrams directly. In addition, cross-domain image transfer is more likely to generate wrong images. After many attempts, we find using bi-level Black/White thresholds to make MRI images in only bi-level B/W result can resolve this issue. Bi-level B/W thresholds, data augmentation, with both cropping and part deformation is employed for the training data.

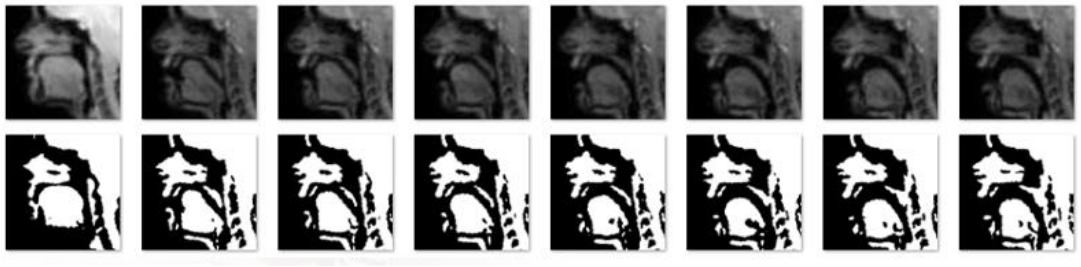

Figure 25 Bi-level B/W threshold is used to create bi-level B/W MRI images

To make our project more generally applicable, we also use YouTube MRI videos as test set. MRI speaking videos were downloaded and the same methods above were applied to the video clips. After testing various images, we determined some issues in the process. Deviations are caused by the differences in the bone and tongue structure of different people. Additionally, many new tongue positions are described in our new data. In our data from *Seeing speech*, we are presented with all the possibilities of typical pronunciation. However, the subjects in the other videos may pronounce in inappropriate tongue positions and sometimes they may still move their tongue in the break which both can cause error outputs.

3.2 Data augmentation

After the pre-treatment, our training dataset can generate relatively high-resolution and accurate speech diagrams from *Seeing speech* MRIs. However, our goal is to generate speech diagrams from different MRI data sources. We also have other free MRI video clips that seem quite different from our dataset. In our observation, different MRI images relating to human speaking or singing are mainly different in image brightness, bone structure and tongue size. As to the difference in brightness and color in bone, our methods of bi-level B/W threshold is a useful approach to mitigate deviations caused by them. For the variation resulting from the facial differences in different people, we use data augmentation to counter these issues.

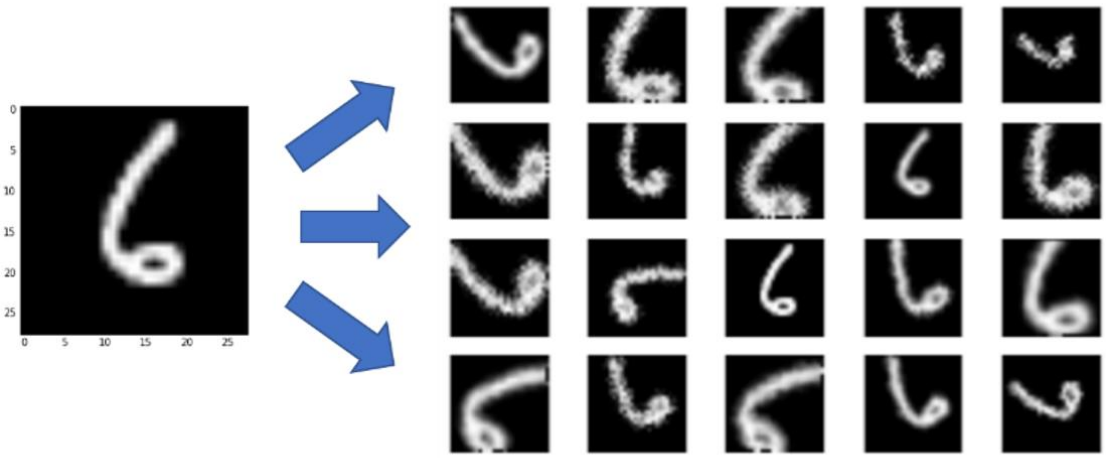

Figure 26 Examples of images sampled from the augmentation pipeline [35]

Data augmentation can assist with the enrichment of our dataset to handle the translation of distinctive MRI images. As shown in Figure 26, typical data augmentation

methods include rotation, Gaussian noise, crop, hue and saturation adjustment, elastic transform, coarse dropout and others.

Our goal is to create a model that can be employed to deal with MRIs from different people, not simply to address those noted in *Seeing speech.* However, typical data augmentation is beneficial for segmentation and recognition, but not for our task. We already have one-by-one matched speech diagrams and MRIs as our data set. If data augmentation methods such as flipping, rotation, cropping or another relevant method is used, the validity of our dataset may be compromised. In other words, we may generate inaccurate speech diagrams when using typical data augmentation methods.

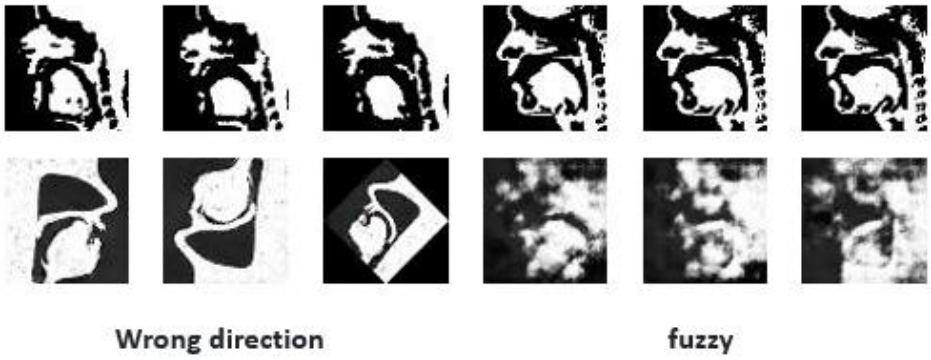

Figure 27 Bad samples of result data caused by wrong use of data augmentation

As it is shown in Figure 26, if we do cropping, rotation, and other common data augmentation methods to training data directly, the generated speech diagram images may be like the modified speech diagram images from the training set.

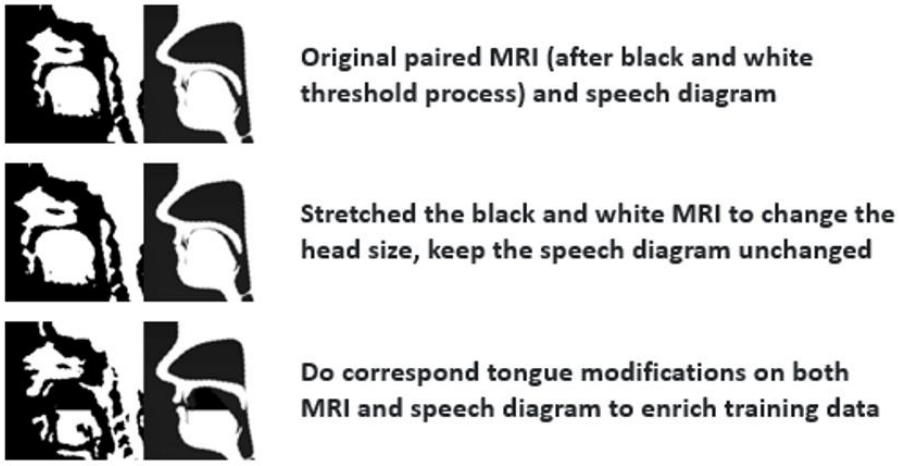

Figure 28 Methods to enrich training dataset inspired by data augmentation

Although typical data augmentation seems to not be useful for our study, we are still inspired by it. When testing MRIs from other data resources, the differences are most prominent on the postures and skeletal features near the tongue. For instance, in some of the external MRIs from other sources, the speakers or singers move their tongues in special ways which are not evident in our training data. To address this, we use MATLAB to execute modifications in the tongue aspect of the training data, both on MRI and speech diagram. As previously mentioned, the variable head size of different people may also cause deviation. We accordingly stretched some copies of MRI images to make the head bigger, meanwhile we used their originally matched speech diagram images, as logically, they were still paired.

3.3 Model

For this topic, we need to generate MRI images from speech diagrams. Since these two kinds of images are not the same in many aspects, we need to find a GAN that can do not only generation but also be used in cross-domain image translation. Cross-domain image translation means the translation from two different kinds of images, for example from human faces to cartoon avatars. So, the relationship between our dataset and result is A: B = C: D, where A is input images, B is the image with the style we want, and both C and D are output images. In this topic, A is a speech diagram, B is the corresponding MRI image and D is the generated MRI image we want. We don't need to generate a speech diagram from MRI in this topic as it seems helpless for language learning, so here we don't need to consider C.

While the organs of human in speech diagram and MRI images are in slightly different positions, we cannot immediately transfer the image style from MRI to speech diagram. Firstly, we tried Cycle GAN [31]. Cycle GAN can be thought of as a paint-style transfer technique that captures the specific features of one data set and figures out how those features translate to other image data sets in the absence of paired training sets. This problem can be broadly referred to as image migration, where images can be transferred from one scene to another. We tried to use Cycle GAN to extract and transfer features between MRI and speech diagram.

Then we tried GAN that can deal with cross-domain image transfer, which means image transfer between two images that are different in shape and kind. Disco GAN [2] is such a tool that can be applied to cross-domain translation. Rather than the overall style transfer mindset in Cycle GAN, Disco GAN focused on the concept that 'one-to-one mapping, rather than many-to-one mapping'. In Disco GAN, the dataset is used after compression. We found that compression is good for efficiency and accuracy, but the result images have a very low resolution.

Dual GAN is also a method to do cross-domain image transformation. It can generate high-resolution images compared to Cycle GAN. However, when tested with PSNR, SSIM and MSE, its results are not accurate compared to ground truth images. On the other hand, the training time is too long.

Finally we choose to build our own model based on Disco GAN. It is more efficient in cross-domain image transfer, but its results are in low resolution. Since low-resolution images cannot be beneficial enough for language learners, we tried some super-resolution methods to solve this issue.

For the traditional single image super-resolution (SISR) problem, the perceived constraints improve the image quality and make the image look clearer, but the hallucinate fake textures and artifacts are produced. After several attempts, we find the idea of Image Super-Resolution by Neural Texture Transfer (SRNTT) metioned in 2.6 can compensate for the details of low-resolution images.

As we explained in 2.6, SRNTT will make the generation results more real but may not be accurate in general cases. For instance, if we use the low-resolution summer street scene picture as input but we use a high-resolution spring street scene picture as reference image. The generated high-resolution street scene picture may make people feel it is in spring. But in our case, both input images and reference images are all same kind. We don't need to use the exact paired MRI as reference image when we do the translation from speech diagram to MRI, we just randomly used original MRI and it still works well. We only need to use the texture to generate MRI that is not blurred or fuzzy.

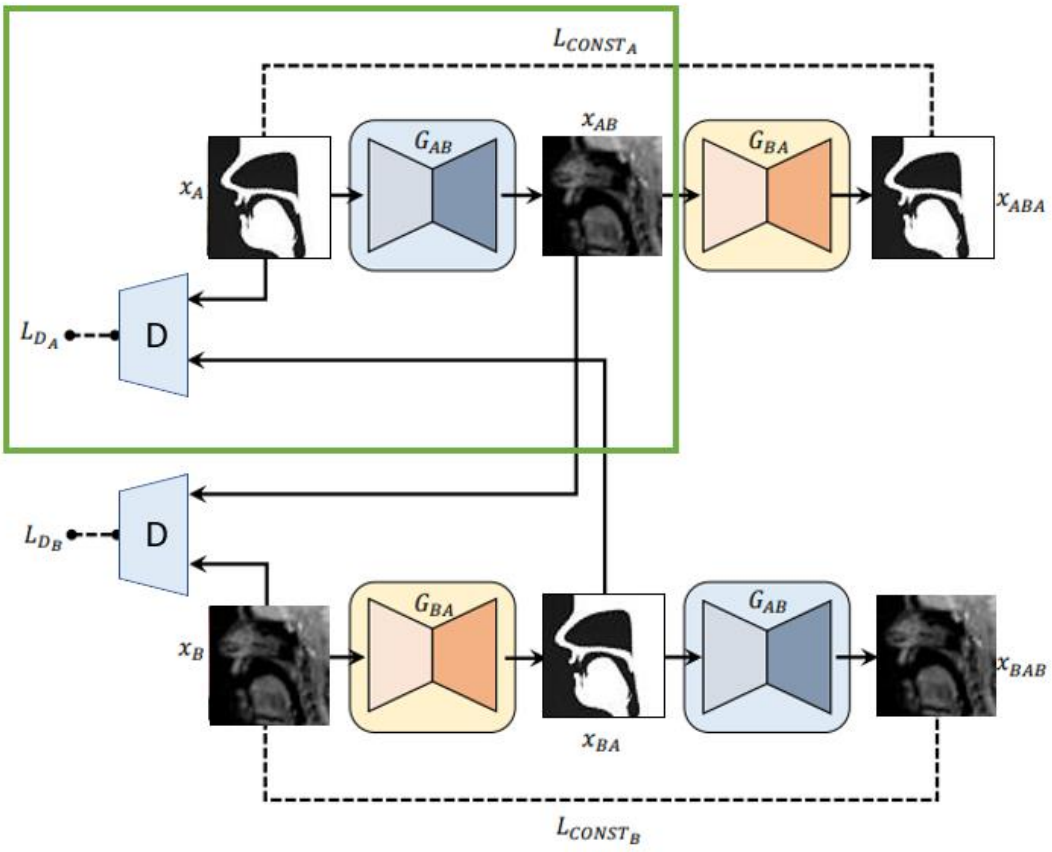

Figure 29 Basic data flow of our task based on Disco GAN[2]

The basic principle of our model is inspired by Disco GAN. The dataflow in the upper part can be described as:

$$X_{AB} = G_{AB}(X_A) \tag{3.1}$$

$$X_{ABA} = G_{BA}(X_{AB}) = G_{BA}(G_{AB}(X_A)) \tag{3.2}$$

where generator G_{AB} translates input speech diagram images X_A to MRI-style images X_B, Then trough generator G_{BA}, X_B is translated to speech diagram images X_{ABA} which matches the input images X_A. The reconstruction loss L_{CONST_A} that compares the input image X_A and the reconstructed image X_{ABA} is defined as:

$$L_{CONST_A} = d(G_{BA}(G_{AB}(X_A)), X_A) \tag{3.3}$$

where d means the distance between the input speech diagram images and the reconstructed speech diagram images. In our method, we also calculate reconstruction loss L_{CONST_A} and L_{CONST_B} in generator loss but we didn't use the same loss function as Disco GAN. A recent study shows Wasserstein GAN (WGAN) loss function has an advantage in accuracy over the traditional GAN loss function [5]. We calculate our generator loss and discriminator loss following the basic principle of WGAN loss. The total objective can be formulated as:

$$\begin{aligned} L_G = {} & \lambda\|X_A - G_{BA}(G_{AB}(X_A))\| + \lambda\|X_B - G_{AB}(G_{BA}(X_B))\| \\ & - D_B(G_{AB}(X_A)) - D_A(G_{BA}(X_B)) \end{aligned} \tag{3.4}$$

$$\begin{aligned} L_D &= L_{D_A} + L_{D_B} \\ &= D_A(G_{BA}(X_B)) - D_A(X_A) + D_B(G_{AB}(X_A)) - D_B(X_B) \end{aligned} \tag{3.5}$$

where λ is a constant default set to100.0.

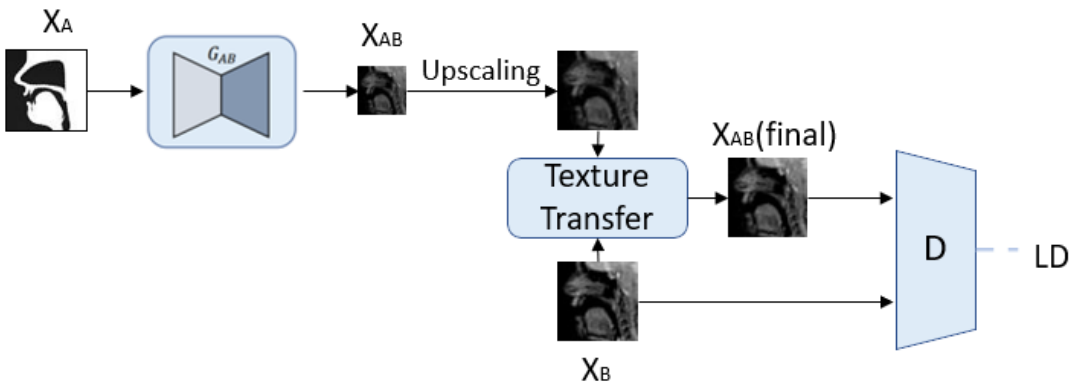

Figure 30 Quarter model of our custom GAN

As we need to apply the super-resolution method SRNTT in our model, we modified the architecture of Disco GAN. The quarter part in the green box in Figure 28 is modified as in Figure 29. We created a generator that produces MRI images X_{AB} (40 × 40) from speech diagram X_A (whose image size is 160 × 160) after scaled down to 40 × 40 for efficiency and accuracy. After upscaling X_{AB} to 160 × 160, we replace the texture from paired MRI image X_B, to produce X_{AB} (final). X_B here is used as a reference picture. We replace the texture in upscaled X_{AB} by corresponding the referenced image, thus we no longer need to deal with blurred MRI.

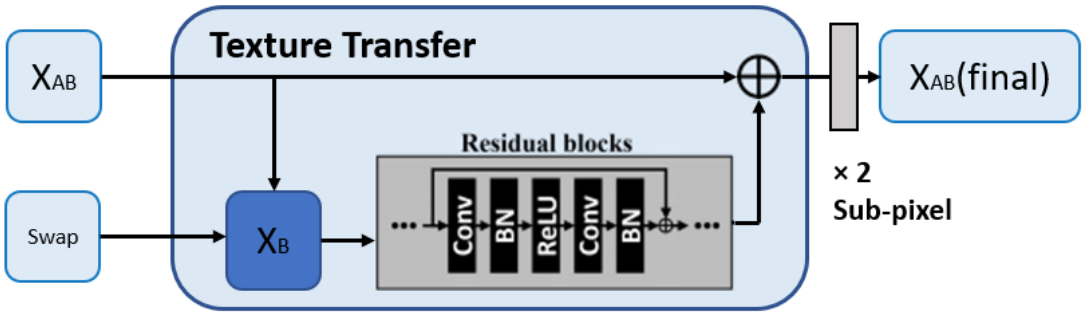

Figure 31 Texture Transfer Model

In the Texture Transfer section, we tried to elucidate multiple swapped texture feature maps and merge them in the generative network. The residual blocks extract related texture from X_B conditioned on X_{AB} and merge it with target content using a set of convolution operations, batch normalization, and relu5 1 layer of VGG19 [36]. At the end of the Texture Transfer opponent, sub-pixel convolution [37] is used for 2× upscaling. It will process twice each time and can be described as:

$$XAB(final) = Res(X_{AB}||X_B)+X_{AB} \quad 3.6$$

where Res(·) is the residual blocks and || is channel-wise concatenation.

4 Experiment and Result

4.1 Environment Setup

The hardware use for the experiments is an Intel-based PC equipped with NVIDIA GPU. The system has a CPU of AMD Ryzen 7 2700x with eight-core processor, 16 GB system memory and 512 GB Solid State Disk. The graphical adapter is NVIDIA GeForce GTX 1080Ti which is a high-end GPU based on Pascal micro-architecture. There is 11 GB GPU-dedicated memory installed on this adapter to enable training of large neural network.

The software framework is Python 2.7, Python 3.6, PyTorch, Numpy/Scipy/Pandas, Progressbar, OpenCV, TensorFlow[38] (version 1.13.1), requests (version 2.21.0), pillow (version 5.4.1), matplotlib (version 3.0.2), CUDA (version 8.0), CuDNN (version 5.1) and maybe other necessary tools. The operation system we use is Linux ubuntu 20.04.

4.2 Experimental Results

For image transfer from MRI to speech diagrams, we used our image processing method which is inspired by data augmentation before training. We didn't immediately use the MRI image to do the translation. Instead, we used bi-level B/W threshold images to generate speech diagrams.

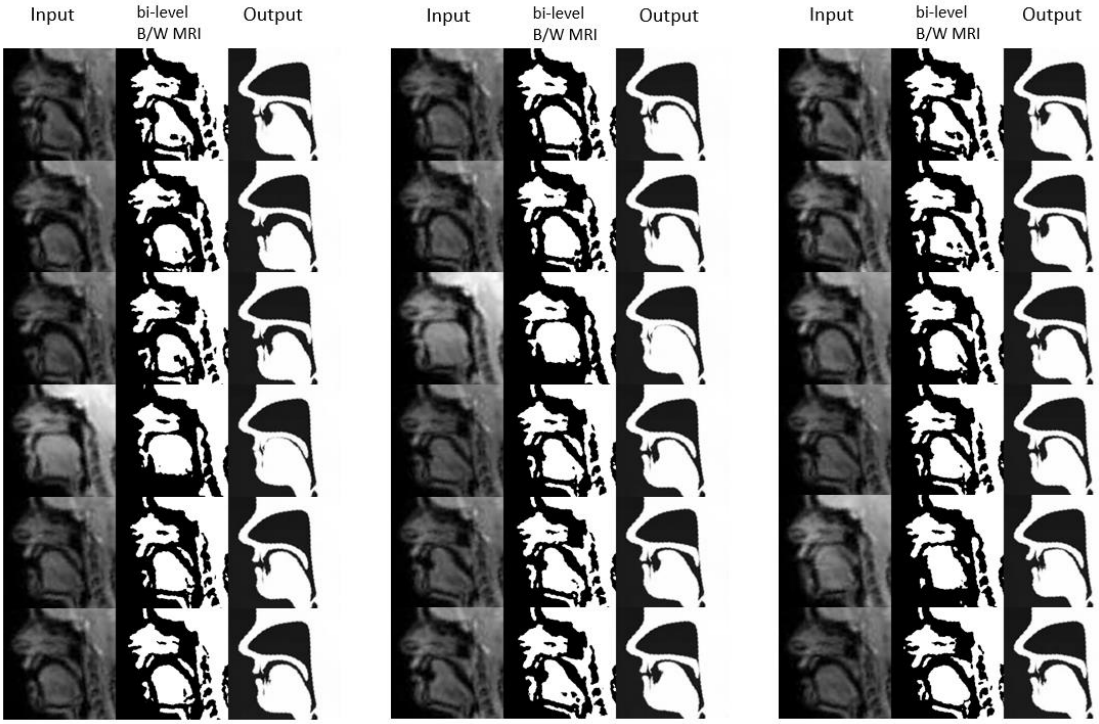

Figure 32 Image transfer from MRI to speech diagrams.

For image transfer from speech diagrams to MRI, we used our custom GAN which is inspired by Disco GAN. But we also modified the architecture by adding super-resolution and changed the loss function to be like the loss function of WGAN. The sum of all these methods enabled us to generate relatively high-resolution speech diagrams with good accuracy and efficiency.

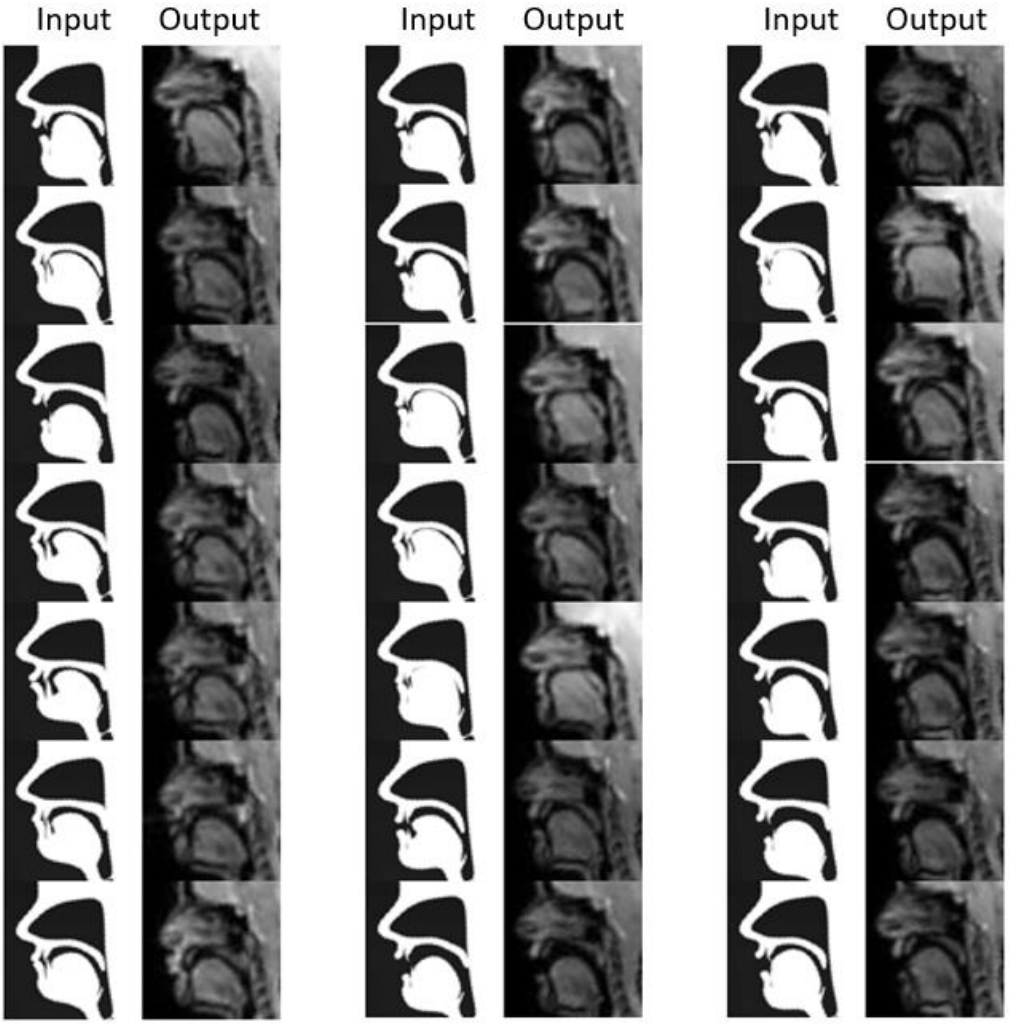

Figure 33 Image transfer from speech diagrams to MRI

4.3 Evaluation Methods

To evaluate the resultant quality of the generated speech diagrams, we used three methods: Mean squared error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity (SSIM). Traditionally the way they are used is by comparing target images with original images. In our case, we use the paired speech diagrams from training data as the ground truth and generated one as images for comparison.

4.4 Ablation Study

In machine learning, ablation study means to remove some feature of the model and see how that affects performance. In our experiments, when we did image transfer from speech diagrams to MRI, the results are possibly generated wrongly. To solve this problem, we tried the loss function inspired by Wasserstein GAN. We expected it to be more accurate than the sigmoid cross-entropy loss from the original GAN. So, we did the ablation study of loss function from WGAN and original GAN by collecting the data of MSE, PSNR and SSIM every 150 epochs. We found that all evaluation data tends to be stable after 750 epochs using WGAN loss. Even after 900 epochs, fluctuations of all the data still existed for the original loss.

Also, we modified the architecture of Disco GAN significantly. In the original model, generated results were compared to low-resolution inputs in the discriminator. In our model, super-resolution is used so we compared higher-resolution generated images with original input images. Using super-resolution can also improve the quality of the generated images according to the evaluation data.

Table 1 Comparison of different loss functions and with/without super-resolution

Methods	MSE	PSNR	SSIM
Original	810.5618	20.4334	0.6074
WGAN loss	756.2541	20.7346	0.6498
SR	723.0297	20.9297	0.6335
WGAN loss & SR	**674.7924**	**21.2296**	**0.6598**

In table 1 we can conclude that WGAN loss with super-resolution gives the most accurate data. By manually checking, we also find there is no significant mismatch between generated data using WGAN loss and the input speech diagrams. Nevertheless, some obviously incorrect generated MRI images can still be found when we use original loss, even after many epochs of training.

4.5 Comparison with other Methods

For image transfer from MRI to speech diagrams, we also tested the same data in Cycle GAN and Disco GAN. Cycle is widely used in image style translation, one well-known application of it is to change horses to zebras or the reverse. Firstly, we use paired MRI and speech diagram to obtain the results. Then MRI images after bi-level B/W threshold are used in training data instead of the origin MRIs. However, Cycle GAN shows strong performance in style transfer but is not quite useful when the architectures of two image groups are different. We can barely find any significant change in the result data compared to bi-level B/W MRI.

Disco GAN can be helpful when dealing with cross-domain image translation. We both use multi-level gray MRI and bi-level black-and-white MRI (bi-level B/W MRI) as training data to generate speech diagrams.

Table 2 Comparison of our method, Cycle GAN [31]. Disco GAN [2] and Dual GAN [3] using multi-level gray MRI and bi-level B/W MRI

	MSE	PSNR	SSIM

Cycle GAN	***multi-level gray MRI***	1973.94	16.5679	0.468
	bi-level B/W MRI	2161.33	16.1740	0.548
Disco GAN	***multi-level gray MRI***	1326.50	18.2942	0.841
	bi-level B/W MRI	1204.36	18.7137	0.835
Dual GAN	***multi-level gray MRI***	989.916	19.5653	0.884
	bi-level B/W MRI	810.174	20.4355	0.876
Ours		**674.792**	**21.2296**	**0.896**

From the data of evaluation methods MSE, PSNR and SSIM, we can see generally that using the bi-level B/W threshold method can make the results more accurate. We listed some of the resultant data generated by different methods who used bi-level B/W threshold before training and testing to make visual comparison. We noticed some interesting points. The generated speech diagrams of Cycle GAN are almost the same as bi-level black-and-white MRI (bi-level B/W MRI), and the evaluation data of its result is quite unstable and varies within a sizable range. The result of Disco GAN has high scores in some evaluation methods. However, by visual assessment we can see in many cases it does not generate speech diagrams with corresponding features as do the MRIs. The blurred and not obviously changed parts of its generated images seem to help to improve the evaluation score. When comparing result data of Dual GAN and MRI, it's easy to see the features of bones and tongues as they are represented well. Our methods to improve the evaluation score are also proven to be effective.

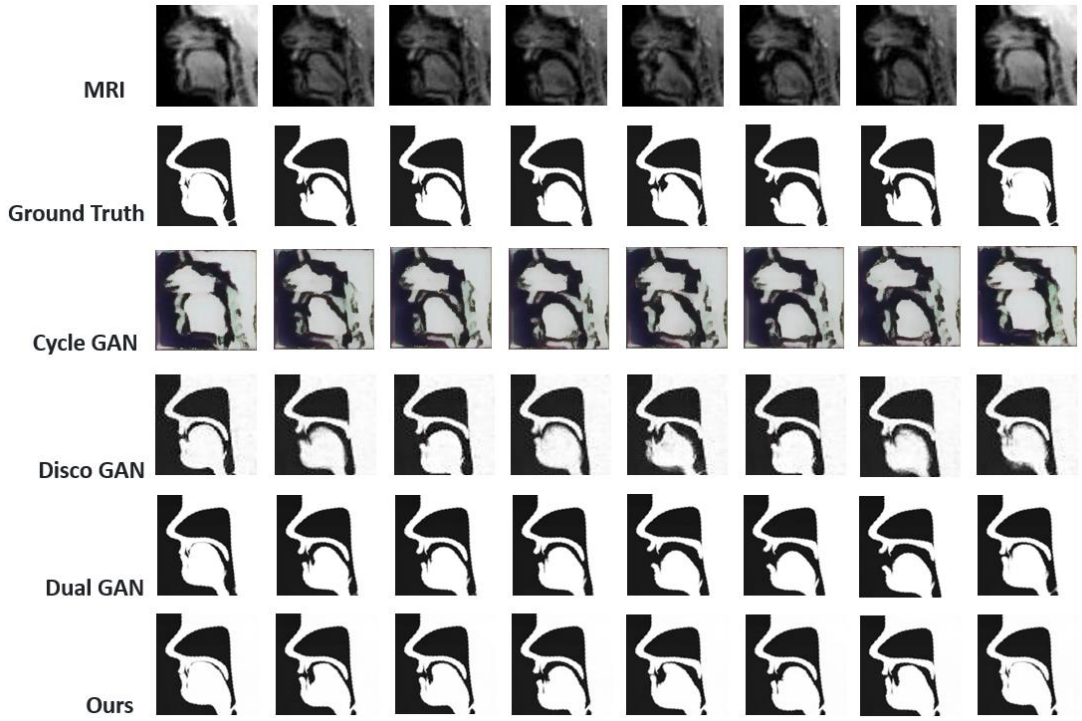

Figure 34 Visual comparison of different methods.

As to image transfer from speech diagrams to MRI, it is believed that Cycle GAN can do this type of task perfectly, so Cycle GAN is also tested. We evaluate results from Cycle GAN, Disco GAN, and our custom GAN by comparing them to the original input MRI data. The original input MRIs are not the ground truth, but they can be the standard to evaluate how accurate the generated MRIs are.

Table 3 Comparison of Disco GAN [2] and our custom GAN in MSE, PSNR and SSIM

Methods	MSE	PSNR	SSIM
Cycle GAN	1773.9386	17.0319	0.1684
Disco GAN	810.5618	20.4334	0.6074
Ours	**674.7924**	**21.2296**	**0.6598**

For image transferring from MRI to speech diagram. Our method is better than other tested methods in MSE, PSNR and SSIM. Because of the limitation we mentioned for Cycle GAN, it shows weak performance in doing this task.

We achieved the relatively satisfying results after 900 epochs of our custom GAN. Compared with other GANs, our custom GAN can generate higher-resolution and more accurate magnetic resonance tongue images from speech diagrams.

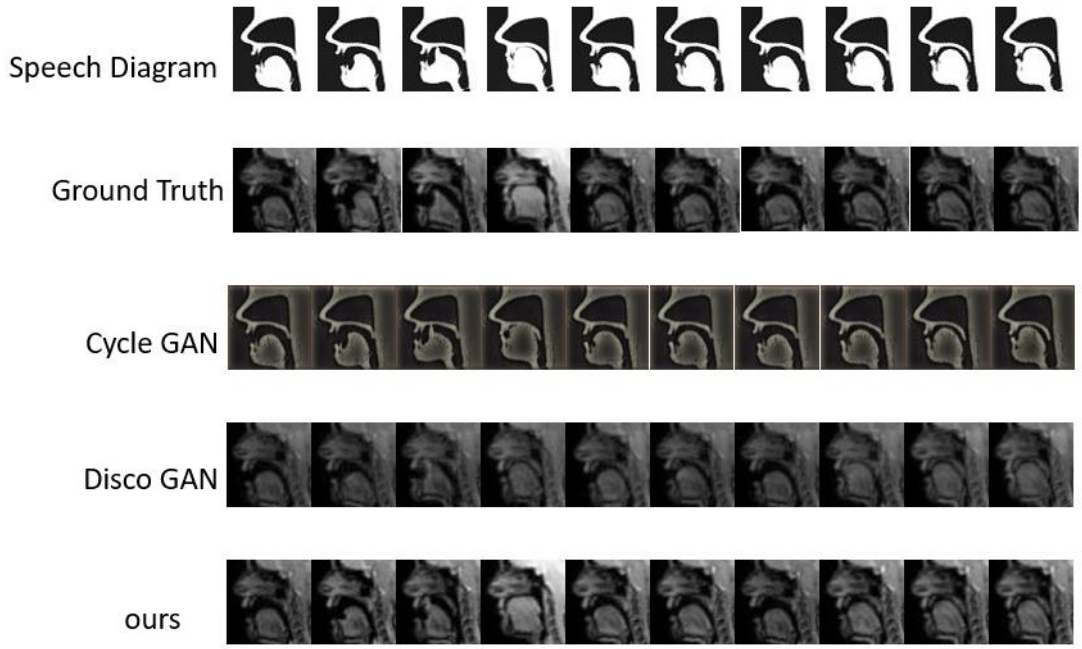

Figure 35 Visual comparison among different related methods

In table 2 it is obvious to see that Cycle GAN is not suitable for this task, and our custom GAN can do a better job in transferring from speech diagrams to MRIs compared to Disco GAN. In figure 31 we can also see our results are far superior in their sharpness and they can be more helpful for further study.

5 Conclusion

Generally, we obtain a positive result for speech diagram generation and MRI generation. We build our custom GAN structure which is inspired by different GANs and different super-resolution methods. This network is built to improve MRI generation from speech diagrams. The basic idea of cross-domain image transformation is used. We also use SRNTT as our super-resolution method. SRNTT extracts features of target images and reference images and replace the textures from the reference images to generated ones to make them in higher resolution. In our case the reference images and the MRIs which we want to generate are very similar so SRNTT seems is relatively the most effective way to be applied. Also, we use the loss function from WGAN to make our result better in accuracy. Secondly, we optimized our training data by using image processing methods and data augmentation methods. They can improve our training data to fasten the training process and to improve our generated data.

Our research can do the transferring between MRI and speech diagram by computer instead of manual work. Also, it has great benefits for medical scientists to do the transformation between MRI and other medical images since we finished the diagrams and MRI translation part and further work is only about translation of speech diagrams and other medical images. Besides, our work can help learners who are willing to improve learning efficiency and overcome learning obstacles through visual

feedback. Our research also contributes to cross-domain image transformation with our methods and evaluation data.

For future work, firstly we will collect more authorized training data from hospitals or research organizations. The data from *Seeing speech* website is too narrow. We need MRIs from different people since their bones and tongues vary in shape and size. Then we need some manual work to make the corresponding speech diagrams to enrich our paired training data. By enough training data, we can make our results more accurate and our project can be more generally used.

Secondly, we also can do the translation from CT or other medical images to MRI or backwards mainly focus on tongue part. We can study the relationship between other medical images and speech diagrams. Since we already can do the transformation between MRI and speech diagram, we can treat speech diagram as a middle process of different transformations between medical images. Speech diagram is suitable to act this role because it has enough necessary information and is relatively easy to study.

6 References

[1] Lawson, E., Stuart-Smith, J., Scobbie, J. M., Nakai, S. (2018). *Seeing speech*: an articulatory web resource for the study of Phonetics. University of Glasgow. 16th August 2020. https://www.seeingspeech.ac.uk/

[2] Kim, T., Cha, M., Kim, H., Lee, J. K., & Kim, J. (2017). Learning to discover cross-domain relations with generative adversarial networks. arXiv preprint arXiv:1703.05192.

[3] Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE international conference on computer vision (pp. 2849-2857).

[4] Zhang, Z., Wang, Z., Lin, Z., & Qi, H. (2019). Image super-resolution by neural texture transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7982-7991).

[5] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein gan. arXiv preprint arXiv:1701.07875.

[6] Suemitsu, A., Dang, J., Ito, T., & Tiede, M. (2015). A real-time articulatory visual feedback approach with target presentation for second language pronunciation learning. The Journal of the Acoustical Society of America, 138(4), EL382-EL387.

[7] Wilhelms-Tricarico, R. (1995). Physiological modeling of speech production: Methods for modeling soft-tissue articulators. The Journal of the Acoustical Society of America, 97(5), 3085-3098.

[8] R. T. Sataloff, (1992). The Human Voice How the voice works was largely unknown until are now improving the care and treatment of the voice, vol. 267, no. December, (pp. 108–115).

[9] Golding, R. P. (1991). Fundamentals of Body CT. Radiology, 181(1), 224-224.

[10] Takano, S., & Honda, K. (2007). An MRI analysis of the extrinsic tongue muscles during vowel production. Speech communication, 49(1), 49-58.

[11] Maricich, S. M., Azizi, P., Jones, J. Y., Morriss, M. C., Hunter, J. V., Smith, E. O., & Miller, G. (2007). Myelination as assessed by conventional MR imaging is normal in young children with idiopathic developmental delay. American journal of neuroradiology, 28(8), 1602-1605.

[12] Karabay, N., Ada, E., İkiz, A. Ö., & Durak, M. G. (2015). Hemoptysis caused by ectopic lingual thyroid. Quantitative imaging in medicine and surgery, 5(3), 480.

[13] McRobbie, D. W., Moore, E. A., Graves, M. J., & Prince, M. R. (2017). MRI from Picture to Proton. Cambridge university press.

[14] Cappa, S. F., Papagno, C., & Vallar, G. (1990). Language and verbal memory after right hemispheric stroke: a clinical-CT scan study. Neuropsychologia, 28(5), 503-509.

[15]. Mozaffari, M. H., Guan, S., Wen, S., Wang, N., & Lee, W. S. (2018). Guided learning of pronunciation by visualizing tongue articulation in ultrasound image sequences. In 2018 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA) (pp. 1-5). IEEE.

[16] Wayland, R., & Li, B. (2005). Training native Chinese and native English listeners to perceive Thai tones. In ISCA workshop on plasticity in speech perception.

[17] Wayland, R. P., & Li, B. (2008). Effects of two training procedures in cross-language perception of tones. Journal of Phonetics, 36(2), 250-267.

[18] Clevert, D. A., Unterthiner, T., & Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.

[19] Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. arXiv preprint arXiv:1412.6806.

[20] In Wikipedia (n.d) Loss function. 13th August 2020. Retrieved from https://en.wikipedia.org/wiki/Loss_function

[21] LeCun, Y. (1998). Efficient backprop In: Neural Networks: Tricks of the Trade, This Book is an Outgrowth of a 1996 NIPS Workshop.

[22] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

[23] Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013, February). On the importance of initialization and momentum in deep learning. In International conference on machine learning (pp. 1139-1147).

[24] Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research, 12(7).

[25] Meyer, D. (2015). Introduction to autoencoders.

[26] Bourlard, H., & Kamp, Y. (1988). Auto-association by multilayer perceptrons and singular value decomposition. Biological cybernetics, 59(4-5), 291-294.

[27] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[28] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).

[29] Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

[30] Emami, H., Aliabadi, M. M., Dong, M., & Chinnam, R. (2020). Spa-gan: Spatial attention gan for image-to-image translation. IEEE Transactions on Multimedia.

[31] Chu, C., Zhmoginov, A., & Sandler, M. (2017). Cyclegan, a master of steganography. arXiv preprint arXiv:1712.02950.

[32] Ward, C. M., Harguess, J., Crabb, B., & Parameswaran, S. (2017, September). Image quality assessment for determining efficacy and limitations of Super-Resolution Convolutional Neural Network (SRCNN). In Applications of Digital Image Processing XL (Vol. 10396, p. 1039605). International Society for Optics and Photonics.

[33] Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), 600-612.

[34] Barratt, S., & Sharma, R. (2018). A note on the inception score. arXiv preprint arXiv:1801.01973.

[35] Van Dyk, D. A., & Meng, X. L. (2001). The art of data augmentation. Journal of Computational and Graphical Statistics, 10(1), 1-50.

[36] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[37] Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., ... & Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1874-1883).

[38] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16) (pp. 265-283).