Enhancing Cross-Modal Translation through Cascade Attention Mechanisms

Submitted in partial fulfillment of the

requirements for the award of

Bachelor of Engineering degree in Computer Science and Engineering

By

YASAR ARAFAT E (Reg.No - 40111467)

RUPESH SURYA B (Reg.No - 40111084)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHISALAI,

CHENNAI - 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Yasar Arafat E (Reg.No - 40111467) and Rupesh Surya B (Reg.No - 40111084)** who carried out the Project Phase-2 entitled **"Enhancing Cross-Modal Translation Through Cascade Attention Mechanisms"** under my supervision from January 2023 to April 2023.

Internal Guide

Dr. J. ALBERT MAYAN, M.E., Ph.D.

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

Submitted for Viva voce Examination held on _____

Internal Examiner **External Examiner**

DECLARATION

I **Yasar Arafat E** hereby declare that the Project Report entitled **Enhancing Cross-Modal Translation Through Cascade Attention Mechanisms** done by me under the guidance of **Dr. J. Albert Mayan** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D.**, **Dean**, School of Computing, **Dr.S.Vigneshwari M.E., Ph.D., and Dr.L.Lakshmanan M.E., Ph.D.,** Heads of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. J. Albert Mayan** for his valuable guidance, suggestions and constant encouragement which paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Computer Science and Engineering who were helpful in many ways for the completion of the project.

ABSTRACT

Since we were babies, we intuitively develop the ability to correlate the input from different cognitive sensors such as vision, audio, and text. However, in machine learning, this cross-modal learning is a nontrivial task because different modalities have no homogeneous properties.

Previous works discover that there should be bridges among different modalities. From a neurology and psychology perspective, humans have the capacity to link one modality with another one, e.g., associating a picture of a bird with the only hearing of its singing and vice versa. Is it possible for machine learning algorithms to recover the scene given the audio signal? In this project, we propose a novel Cascade Attention-Guided Residue GAN (CAR-GAN), aiming at reconstructing the scenes given the corresponding audio signals.

Particularly, we present a residue module to mitigate the gap between different modalities progressively. Moreover, a cascade attention guided network with a novel classification loss function is designed to tackle the cross-modal learning task.

Our model keeps consistency in the high-level semantic label domain and is able to balance two different modalities. The experimental results demonstrate that our model achieves the state-of-the-art cross-modal audio-visual generation on the challenging Sub-URMP dataset.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Cross-modal learning involves multiple modalities, aims at learning knowledge from one modality to facilitate the tasks (e.g., retrieval and generation) from another correlated modality. Cross-modal learning gains long-lasting interest in multimedia. Recently, with the increasing popularity of Generative Adversarial Networks (GANs) [1], cross-modal research is not only limited to retrieval [2], [3] but also makes the cross-modal generation possible, such as text-to-image [4], [5], image-to-image [6] – [9], story visualization and generation [10], [11]. Recently, radio signals [12] have also been successfully applied to human pose prediction. The radio signals which are a form of waves that are robust to occlusions so that it can predict human poses behind the wall. Another special wave is an audio signal, which has also been explored to reconstruct the scene [13]– [17]. Generating images from audios using GANs was first described by Chen et al. [13] where they introduce a conditional GANs (cGANs) [18] model to tackle the problem. Later, Hao et al. presented the Cross-Modal Cycle GAN (CMCGAN) [14] to solve the cross-modal visual-audio mutual generation problem. Although this paper conducts an interesting exploration, we still observe unsatisfactory artifacts and missing contents in the generated images, which are due to several reasons. First, even if previous works [13], [14] showed there were truly some connections between audio and visual modalities, there is still a huge gap between different modalities, e.g., the sound of the wind blowing trees and the image of shaking leaves. Thus, without prior knowledge about this scenario, it is hard to associate them together. Second, a random latent vector was employed to assist the learning process. They tried to represent the properties of the input audios accompanied by some manually defined random latent vectors. However, we argue that these latent vectors cannot represent the information of the audios accurately since they are random Gaussian noises, i.e., they are not directly withdrawn from the audios. Consequently, we avoid employing random latent vectors in our designed model. Based on the above observations, in this paper, we propose a novel Cascade Attention-Guided Residue GAN (CAR-GAN)

to handle the audio-to-image translation task.

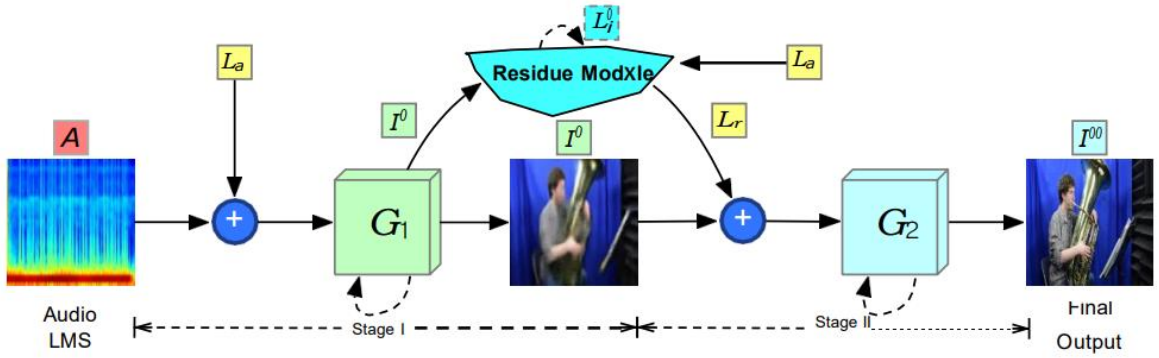


Fig. 1.1 Overview of CAR-GAN framework. Generator G1 takes audio LMS (A) and its class label (La) as inputs to generate coarse image I ′ . Generator G2 takes I ′ and the output of our specific-designed residue module (Lr) as inputs to synthesize fine-grained image I ″ . Note that to detect the most distinguished part in different modalities, we introduce the self-attention mechanism to both generators. Two generators of different stages are jointly optimized in an end- to-end fashion that aims at enjoying the mutually improved benefits from different modalities, i.e., audio and image. Ⓧ denotes channel-wise concatenation.

The proposed CAR-GAN contains two-generation stages and the overall framework of CAR-GAN is depicted in Fig. 1, in the first stage, the Log-amplitude of Mel-Spectrum (LMS) [19] image A (a representation of the raw audio) concatenated with La (one-hot class label for the audio) is fed into the first self-attention guided generator (G1) and G1 outputs a coarse result I ′. Different from previous works [13], [14], we deprecate the latent vector in our model. Note that to efficiently model relationships between widely separated spatial regions, we introduce self-attention to both generator and discriminator. The coarse output I ′ from the first-generation network is taken as input to the proposed residue module to obtain the corresponding class label vector L′ i, which is a high-level semantic class label for the coarse output I ′ produced by the embedded classifier in our proposed residue module. The label Li reflects how realistic the first generator is by measuring the embedding in the high-level class label space. Then we subtract L′ i from label La to obtain the residue class label vector Lr. Lr reflects the discrepancy between the generated image I ′ and the audio in the semantic label domain based on the coarse output I ′, which is corresponding to the discrepancy between L′ i and La. By taking means of Lr, our proposed model makes compensation for the information lost in the first stage and generates more realistic results in the second stage. Next, the coarse output I ′ from the first stage, together with the residue class label Lr, are input into the second stage network and generate more fine-grained final results. The intuition behind the

residue module is that the second generator G2 can flexibly preserve the similarities between audio and image space and only model the differences when it is necessary, which can be regarded as a progressive generation strategy. Finally, to optimize the proposed CAR-GAN in an end-to-end fashion, the cascaded residue classification loss is further used to generate more realistic images and preserve the consistency between two stages. It is worth noting that, the classifier in our proposed residue module is pre-trained with the real images from the Sub-URMP dataset. During the training of our CAR-GAN model, the classifier parameters are fixed, and only the gradients of the synthesized images will be backpropagated to guide the generator to synthesize images with correct semantic labels. In this way, the label consistency between the synthesized images and the audio labels could be preserved. Through extensive experimental evaluations, we demonstrate that CAR-GAN produces better results than the baselines such as S2IC [13] and CMCGAN [14]. Overall, the contributions of this paper are summarized as follows:

• A novel Cascade Attention-Guided Residue GAN framework (CAR-GAN) for the cross-modal audio-to-image translation task is proposed. It explores cascaded attention guidance with a coarse-to-fine generation, aims at producing a more detailed synthesis from the jointly learned representation of both audio and image spaces. • A novel residue module is presented, which is utilized to smooth the gaps between different modalities at class label space and is able to find correlations between different modalities. We also propose a new cascaded residue classification loss for more robust optimization. It not only helps the model generate more realistic images but also keeps the consistency between the two stages' generation processes.

• Qualitative and quantitative results demonstrate the effectiveness of the proposed CAR-GAN on the cross-modal audio-to-image translation task, and show state-of-the-art performance on the challenging Sub-URMP dataset [20] with remarkable improvements.

CHAPTER 2

RELATED WORK

Generative adversarial Networks (GANs) is proposed by Goodfellow et al. [1], complemented with adversarial method. A vanilla GAN is composed of a generator and a discriminator. The discriminator is trying to discriminate whether an image is real or fake. Conversely, the generator is to learn to output images that can fake the discriminator. Since the GANs appeared, plenty of works such as [6], [11], [21], [22] on computer vision are based on GANs. With the success of GANs, conditional GANs encode additional information as a reference into the GAN framework which will make sure the generator can run more straightforward to the target. cGANs have achieved remarkable results in image-to-image translation [6], [7], [23], [24], super-resolution image generation [25], [26] and style transfer [27], [28]. Image-to-Image Translation adopts input-output data to learn a translation mapping between input and output domains. For instance, Isola et al. propose Pix2Pix [6], which is a generalpurpose solution to image-to-image translation problems. To further improve the quality of the generated images, works such as [29]–[31] try to employ the attention mechanism to force the generator to pay more attention to the distinguished content between different input and output domains. In this paper, we embed the proposed attention mechanism into our cross-modal GAN model, which allows the generator to effectively pay attention to the most distinguished representations between audio and image modalities. Moreover, previous works such as [32], [33] generate images using residual images La L 0 i Residue ModXle A I 0 I 0 Lr I 00 + G1 + G2 Audio LMS Stage I Stage II Final Output La which is different from ours, we employ the residual class label to guide the generator for producing photo-realistic images. In this way, the generator only needs to focus on the high-level difference between the audio representation and image representation. Cross-Modal Learning represents any kind of learning that involves information obtained from more than one modality. Earlier work such as [34]–[37] show cross-modal perception phenomena from the perspective of the neurological and psychological field. They try to figure out the mutual relation between auditory and visual information from a neurologist's or psychologist's view. Later, cross-modal multimedia retrieval starts booming since the revolution of multimedia technology. Works such as [2], [3] take advantage of cross-modal learning to help retrieving. Afterwards, cross-modal learning is popular together with generative models, e.g., Variational Autoencoders (VAEs) [38] and GANs [4], [39], [40]. Audio-to-Image Translation. They try to solve the problem by making use of the cycle of the encoders and decoders. Besides, existing methods [13], [14], [41] on audio-to-image translation take the advantage of latent Gaussian vector, where they design front convolution neural networks encoder to extract feature maps out of input audios. Later, the extracted feature maps are concatenated with the latent vector as new feature maps, the combined feature maps are fed into the generator to produce corresponding images, i.e., the latent vector plays an important role in this translation. However, different from these existing methods, we propose replacing the latent vector with the proposed residue class-label since it contains more meaningful representations between different modalities.

CHAPTER 3

CASCADE ATTENTION GUIDED RESIDUE LEARNING GAN

In this section, we describe our proposed Cascade AttentionGuided Residue GAN (CAR-GAN) framework for cross-modal translation in detail. We start with the model formulation and then introduce the proposed objectives. Finally, we present the implementation details including network architecture and training procedure. The overall framework of the proposed CAR-GAN is illustrated in Fig. 1. In stage one, we present a cascade attention guided generation sub-network, which utilizes both the audio signal A and its class label La as inputs to generate an image. The generated image I ′ is further fed into the proposed residue module to obtain the corresponding image class label L_i'.

Next, we calculate the residual cross-modal label Lr between the audio label La and the image label L_i'. Lr reflects the distance between the generated images and the real images in the semantic domain.

In stage two, the coarse synthesis I′ and the residual cross-modal label Lr are combined as the input. In this way, the semantic difference between the generated images and audio signals, Lr can be employed to guide the generator to further refine the generated image L_i'. As a result, the cross-modality semantic distance could be further reduced after the refinement.

3.1 CASCADE ATTENTION GUIDED GENERATION

Class-Label Guided Generation with Self-Attention. Translating audio into an image is an extremely challenging task since it is difficult to tell any relationship between audio and image modalities directly. To handle this challenge, previous works such as S2IC [13] and CMCGAN [14] tried to employ a random Gaussian noise vector as input to guide the generator to produce a synthetic image. We argue that the Gaussian noise vector will introduce some errors misguiding the generator. Different from them, we input a more accurate audio classlabel into the generator similar to [23], [42]. Specifically, as shown in Fig. 1, we first replicate La spatially along with both height and weight dimension and then perform channel-wise concatenation with input LMS A from audio space. Finally, we input them into the first generator G1 and synthesize its corresponding coarse image I ′ as I ′ = G1(A, La). In this way, the audio class-label La provides stronger supervision to guide cross-modal translation in the deep network. Moreover, to force the generators to pay more attention to the most distinguished content between

different modalities, we further introduce the self-attention mechanism into the generators. Zhang et al. [30] proposed the Self-Attention Generative Adversarial Network (SAGAN) for image generation tasks. Differently, in this paper, we propose a self-attention imageto-image translation network which allows long-range dependency modeling for cross-modal image translation task with drastic domain change. Once the generators know which part, they should pay attention to, the next goal is to generate images with more fine-grained details. Therefore, we cascade two generators and train them simultaneously. Coarse-to-Fine Cascade Generation. Due to the complexity of the cross-modal audio-to-image translation task, we observe that the first stage generator G1 only outputs a coarse synthesis with blurred artifacts, missing content, and high pixel-level dis-similarity. This thus inspires us to explore a cascade generation strategy to boost the synthesis performance from the coarse predictions. The **Coarse-to-fine strategy** has been used in different computer vision applications achieving promising performances, such as semantic segmentation [43] and object detection [44], [45]. In this paper, we adapt the coarse-to-fine strategy to handle a more challenging audio-to-image translation task. We observe significant improvement using the proposed cascade coarse-to-fine strategy, which is illustrated in the experimental section.

3.2 CROSS-MODAL RESIDUE LABEL GUIDED GENERATION

The overview of our proposed residue cross-modal label guided generation module is shown in Fig. 2. This module consists of a pre-trained classifier to preserve the cross-modal label cycle consistency and a cross-modal residue label guided generation sub-network. Cross-Modal Label Cycle Consistency. Based on the theory [23], we expand it into our label consistency method. The coarse output I ′ of stage I is fed into the classifier C to generate an image classification label L′ i . To further reduce the space of possible mismatch between audio and image modalities, we hypothesize that the learned mapping functions should be cycle-consistent in cross-modal translation. For the audio class label La, the translation cycle should be able to bring it quite close to the image label L′ i , i.e., G1(A, La) -> I ′ -> C(I ′) -> L′ i =La.

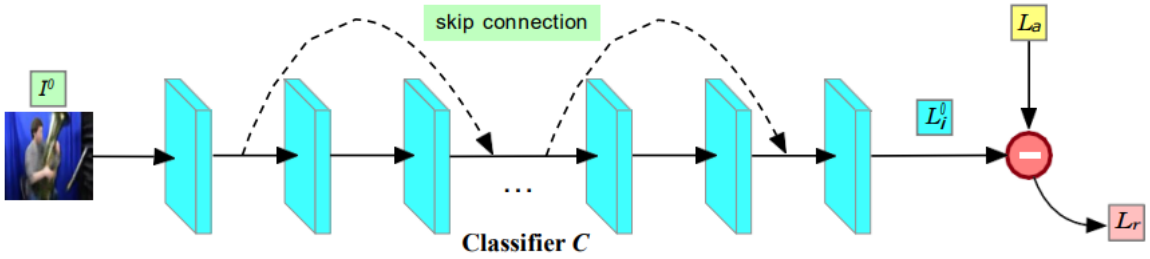


Fig. 3.1 Residue Module: I ′ denotes the output of the first generator G1. The output of the classifier C is L ′ i , which is the predicted label of I ′ . La is the label of the audio, Lr denotes the difference between two different modalities, i.e., audio label La and image label L ′ i. Ⓧ- denotes channel-wise subtraction.

We name this as cross-modal label cycle consistency since L′ i is a label representation of the coarse output I ′ in the image modality and La is a label representation of the audio A in the audio modality. Note that the proposed cross-modal cycle consistency is different from the cycle consistency in CycleGAN [7] which adapts the cycleconsistency between the input image and the reconstruction image in the image space, in this paper, we employ making two different modalities cycle consistent in the class-label space. Cross-Modal Residue Label Guided Generation. Previous works have shown that residual images can be effectively learned and used for image generation task. For instance, Shen and Liu [32] used the learned residual image as the difference between images before and after the face attribute manipulation. Zhao et al. [33] trained networks to learn residual motion between the current and future frames for the image-to-video generation task, which avoids learning motionirrelevant details. Instead of manipulating the whole image, both approaches proposed to learn the residual images. In this way, the manipulation can be operated efficiently with modest pixel modification. However, in the paper, we propose the residue label rather than the residue image for the cross-modal image translation task. Specifically, we first obtain the residue label Lr between the image label L′ i and the audio label La by calculating Lr = La L′ i . Then we intend to generate the missing information Lr in the second generator stage, which can be expressed as, I ″ = G2(I ′ , Lr) = G2(G1(A, La), Lr). (1)

In this way, the generation process can be operated efficiently with modest pixel modification, i.e., the generator G2 can flexibly preserve the similarities between the audio and image representations, and only model the differences between them.

3.3 LABEL CONSISTENCY: BACKPROPAGATING VIA THE CLASSIFIER

A vanilla cGAN conducts backpropagation mainly determined by the discriminator. The discriminator judges from the image-level information, but not the label-level semantic information. We argue that different but corresponding modalities can match with the same semantic label. Therefore, apart from performing backpropagation from the discriminator, we also backpropagate from the label classifier to make sure the generated images belong to the same label domain with input audios. During the training, the corresponding label, i.e., the instrument type of the input audio signals, is fed to the classifier. When we update the model during backpropagation, the parameters of the pre-trained classifier are fixed. Only the gradients of input images are passed back such that the images can be revised accordingly to match its semantic label. In this way, label consistency can be guaranteed. The classifier is re-trained using a pre-trained model using ImageNet [46]. The backpropagating path is described in Fig. 3. During the backpropagation of the classifier, we design a joint loss C for our two-stage generations. C is composed of two parts: the classification loss of stage I and stage II, LC = L(I ′ , I ″) =λ 'L(I ′) + λ "L(I ″), (2)
where is the cross-entropy loss function. λI' and λI" are coefficients to control the relative importance of the two objectives.

3.4 UNIVERSAL OPTIMIZATION OBJECTIVE

Adversarial Loss. Our adversarial loss is composed of two parts since we adapt two stages generation. During the stage I, the adversarial loss of discriminator D for differentiating generated audio-image pairs [A, I ′] from real audio-image pairs [A, I] is formulated as following: LcG1 (A, I ′) = EA,I [log D(A, I)] + EA,I' log(1 − D(A, I ′)) , (3)
where I is the ground truth image. During the stage II, the adversarial loss of discriminator D for differentiating generated audio-image pairs [A, I ″] from real audio-image pairs [A, I] is formulated as following:
LcG2 (A, I ″) = EA,I [log D(A, I)] + EA,I" log(1 − D(A, I ″)) . (4)
The above two adversarial losses both target to reduce the disagreements with ground truth images and generate more realistic synthesized images. Therefore,

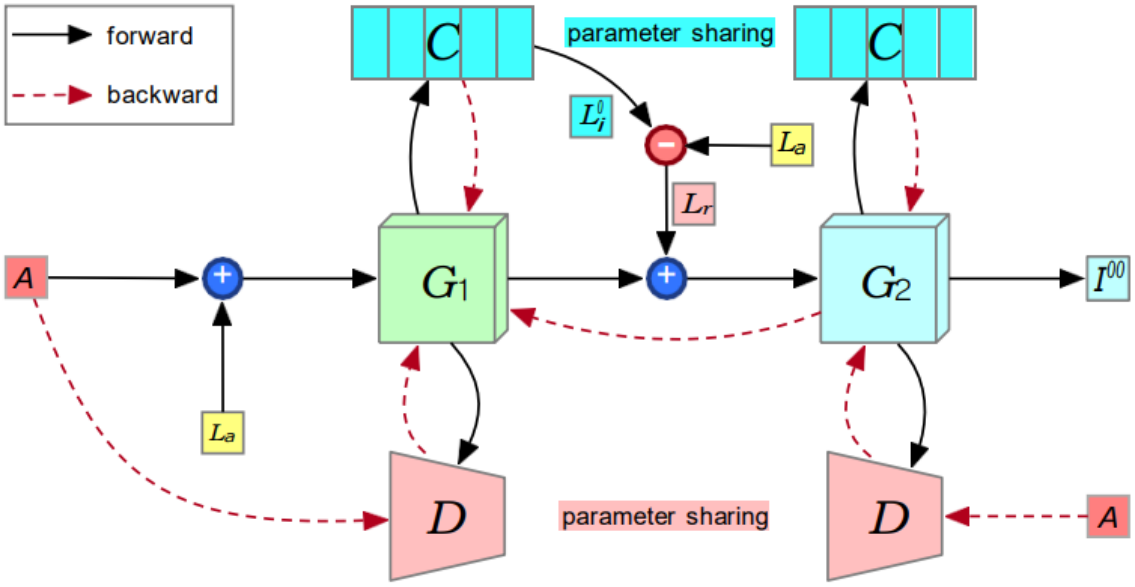


Fig. 3.2 The backpropagation path of the proposed CAR-GAN. The solid line denotes the path for the forward process, the red dashed line denotes the backpropagation path. We use the same pretrained classifier C twice in our model, so does the discriminator D but without pretrained. I ″ denotes the final generated cross-modal image. ⓧ+ , ⓧ- denote channel-wise concatenation and subtraction, respectively.

our adversarial loss is the total sum of the two stages: LcG = λG1 LcG1 (A, I ′) + λG2 LcG2 (A, I ″) (5)

Universal Loss. Besides the adversarial loss cG, we also introduce the classification loss C and L1 loss function for better optimizing our CAR-GAN. Our final loss function is a combination of the three losses. min max = λc C + λcG cG + λL1 L1, (6) 1,G2,C {D} where L1 = L1(I, I ′) + L1(I, I ″). λc, λcG and λL1 denote the trade-off parameters to control the significance of its corresponding loss function, respectively. Our model is trying to balance the min-max problem while training.

3.5 IMPLEMENTATION DETAILS

Network Architecture. Inspired by the work of Isola et al. [6], we employ U-Net [47] as the backbone of our generators G1 and G2. U-Net is a Convolution Neural Network (CNN) architecture with skip connections between a down-sampling encoder and an up-sampling decoder, and it retains complex texture information of the input. We share the same network architecture in both generators. The convolutions of down sampling layers and up-sampling layers are 4 4 kernel with

stride 2 and padding 1. The filters in attention convolution layers are 1 1 with stride 1. For the discriminator D, we adapt PatchGAN as in [6], [7]. The kernel size of the attention convolution layers is also 1 1. Both the generators and discriminator have attention layers before the last two convolution layers. The other convolution layers in the discriminator have a kernel size of 4 4 kernel with stride 2 and padding 1. Batch normalization is used in our model. As for the classifier, we employ ResNet50 [48] architecture which is pre-trained on the ImageNet. Then we add a fully connected layer at the end of the network and conducted transfer learning in the Sub-URMP dataset for high classification accuracy. The classifier is fixed while training our model. Training Details. First, we employ preprocessing for every audio, where the audios are converted from waveform pattern to LMS pattern, which is a frequency warping pattern that allows for better representation of audio clip. After preprocessing all audios, we then input the LMS pattern into our proposed CAR-GAN. Moreover, the proposed CAR-GAN is trained and optimized in an end-to-end style. C is pre-trained and fixed while training. We first train G1, G2 with D fixed, and update parameters of G1 and G2 by the sum of gradients from C's and D's backpropagation, and then we train D with G1 and G2 fixed, but the backpropagation of C has no influence on the optimization of D, i.e., the optimization is only determined by G1 and G2. We apply Adam algorithm [49] for optimizing both the generators (G1, G2) and discriminator (D) jointly. The betas of the Adam algorithm set to 0.9 and 0.999, respectively. Weights are initialized from a Gaussian distribution with standard deviation 0.2 and mean 0.

CHAPTER 4
EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets. Following [13], we adapt the widely used SubURMP (University of Rochester Musical Performance) dataset [20] to evaluate the proposed model. This dataset consists of 17,555 pairs of audios and images and has 13 kinds of instruments played by different people. It maps an image to half-second long audio, and the image is the first frame of the half-second long audio. Parameter Settings. We resize images to 256 256 resolutions as inputs. We implement with Pytorch and the experiments are running at 4 Nvidia GeForce GTX 1080 Ti GPUs with batch size 64. We set the learning rate to 0.0008, and stop our training at the epoch of 200. Both λI' and λI'' in Eq. (2) are set to 0.5. We set λG1 , λG2 in Eq. (5), and λc, λcGAN in Eq. (6) all equal to 1, and λL1 equal to 100 in Eq. (6). Evaluation Metrics. To compare with previous work [13], [14], we employ the classification accuracy as the metric, the only metric used in the previous two papers. The way we measure our model is that we first train a model with 99.56% classification accuracy using ResNet50 trained on the Sub-URMP dataset. Specifically, the 99.56% classification accuracy is obtained by training and testing on both the training and testing set.

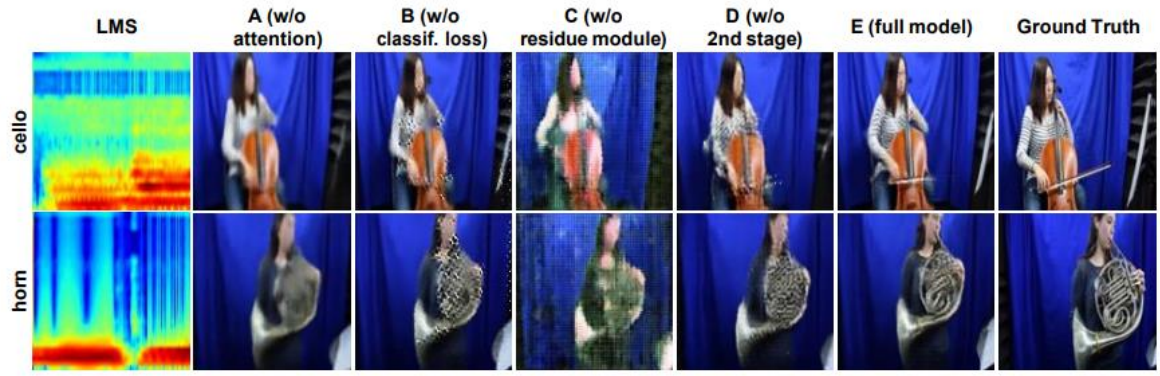


Fig. 4.1 Ablation study: synthesized images by different models of the proposed CAR-GAN. LMS represents the LMS of the input audios.

TABLE I
RESULTS OF THE PROPOSED CAR-GAN FOR FID AND IS METRICS.

Baseline	A	B	C	D	E	GT
FID	279.3022	332.1574	380.1104	307.3725	**207.3734**	-
IS	3.2221	3.4337	2.0699	3.7215	**3.8180**	4.7552

TABLE II
THE CLASSIFICATION ACCURACY OF DIFFERENT METHODS.

Method	Accuracy	
	Training	Testing
S2IC	0.8737	0.7556
CMCGAN	0.9105	0.7661
Ours	**0.9954**	**0.9068**

Then we test the pre-trained ResNet50 on our generated results. Following the same train/test split employed in Table 2 of [13], our model is trained using the Sub-URMP dataset. The intuition behind this is that if the generated images are realistic, the classifier trained on the real images will also achieve decent accuracy on the generated images during the testing stage. In addition to classification accuracy, we also employ Fre´chet Inception Distance (FID) [50] and Inception Score (IS) [51] metrics to further evaluate the fidelity of the generated images. Due to the lack of a pretrained model and released code of previous work, we are not able to get FID and IS of their works.

4.2 EXPERIMENTAL RESULTS

Settings of Ablation Study. We perform ablation studies on the proposed CAR-GAN. We break down our model and assemble it into five different models. The following five models share a similar backbone, but a particular part is abandoned. Model A avoids using attention guided generations. Model B drops out the c loss. The residue module is taken out from model C and Lr is replaced by La. Model D is running without stage II (no second generator). And the last model E is our fully proposed model. Fig. 4 shows the corresponding crossmodal generated images of different models. Table I depicts how models perform based on the metrics we employed.

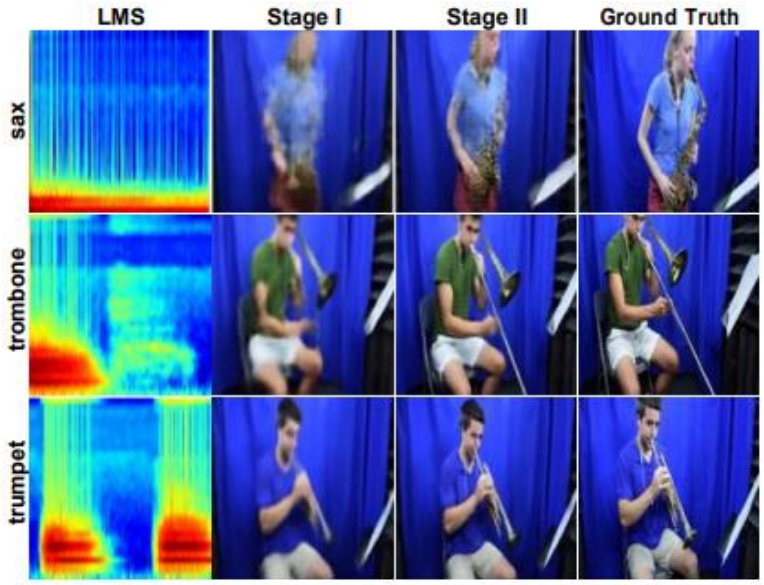


Fig. 4.2 Generated images of different stages of our model.

Influence of Attention Mechanism. Compared with our proposed full model E, model A which avoids the attention mechanism performs slightly worse. The results of model A shows the overall contour of images, but some details are left out. That is, the attention mechanism does enhance the representation ability of our model.

Influence of Classification Loss. If we add our proposed classification loss into the model, we make an improvement by 37.56% in FID and 11.19% in IS since the classification loss c awards the generators strong guidance towards the ground truth. This tells generators from an overall classification view. Therefore, the generators know the appropriate direction to go from the label domain.

Influence of Residue Module. With our residue module, we achieve an amazing improvement on FID by 45.44% and on IS by 84.45%. Our residue module supplements the missing information during the generation of Stage I, making that the generated images belong to the same domain and keep balance in the label domain between inputs and outputs.

Influence of Two-Stage Generation. Our two stages of generators lead to the improvement of 32.53% in FID and 2.59% in IS. The single generator has weak representation ability for a complicated cross-modal generation. Thus, the union of two or more generators can progressively improve representation ability and performance. To visualize the influence of the twostage generation, we display synthesized images by different stages in Fig. 5. The generated images by the

second stage have more fine-grained details and are more realistic, i.e., it certifies that the two-stage generation is beneficial for the whole generation procedure.

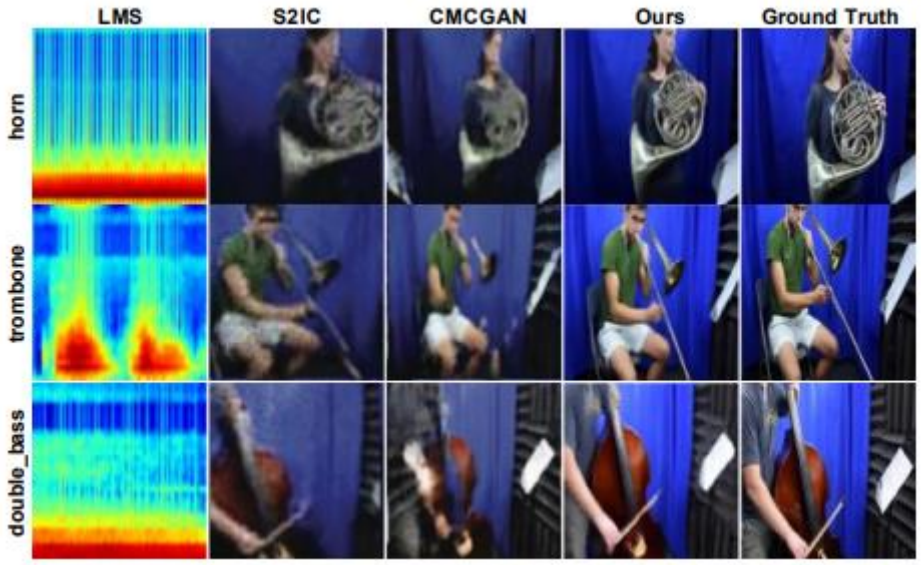


Fig. 4.3 Synthesized images of different methods on the Sub-URMP dataset.

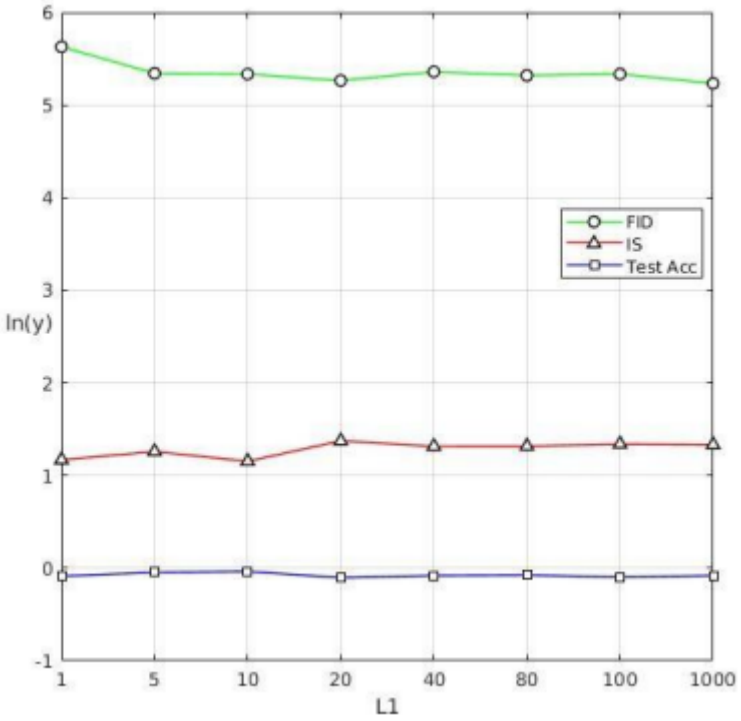


Fig. 4.4 Impact of L1 regularizer. y denotes the performance for each metric. The figure shows how the metrics' value changing with different weight of L1 regularizer in loss function. We adapt ln function on y for better visualization.

Impact of L1 regularizer. To show the model is not overfitting by L1 loss, a more detailed experiment is performed to evaluate the impact of the structural L1 regularizer for the generated images. To better show the different results, we plot the trajectory of the metrics' value in Fig. 7.

State-of-the-art Comparisons. We show the quantitative and qualitative results with comparison methods in Table II and Fig. 6. We make an improvement in training accuracy by 13.93%, 9.32% compared to S2IC [13] and CMCGAN [14], respectively. As for the testing, we achieve an increase of 20.01% and 18.37% compared with S2IC and CMCGAN. Furthermore, we produce more realistic and detailed images compared with the previous methods as illustrated in Fig. 6.

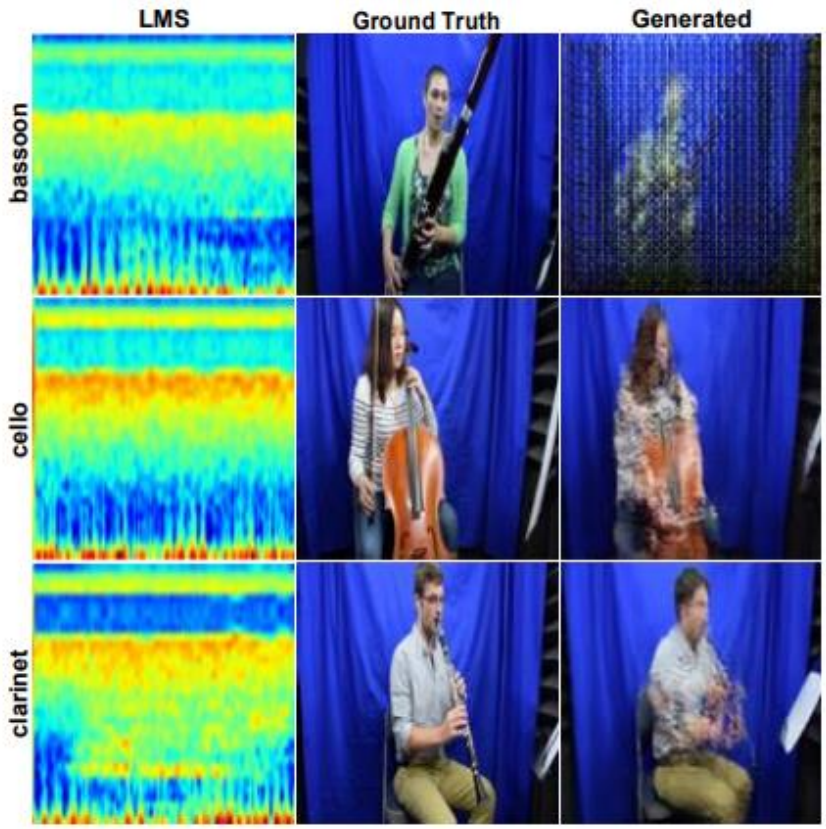


Fig. 4.5 Failure cases during inference on the Sub-URMP dataset.

Failure Cases and Analysis. During our experiments, we find there are some synthesized images like Fig. 8 which are randomly combined by learned features. Moreover, when we carefully look into these cases, we finally find out that the corresponding inputs of these failure cases are more like noises that are randomly distributed, and the GT images show people mostly hold the instrument still and wait, that is, there is no sound making by instruments, the audios mainly consist of background and other noises.

CHAPTER 5
CONCLUSION

In this project, we design a novel Cascade Attention Guided Residue Learning GAN (CAR-GAN) to solve the challenging cross-modal audio-to-image translation task. Particularly, it employs cascaded attention guidance and a coarse-to-fine generation strategy. A novel residue learning model is also proposed to tackle the cross-modal class-label dis-match problem between audio and image modality. By introducing the residue module, generators learn to produce residue features between two stages, which pushes the output closer to its corresponding real image in a high-level semantic space. Finally, the proposed joint classification loss facilitates the model generation and keeps consistency in the label domain. Experimental results show the state-of-the-art performance on the cross-modal

REFERENCES

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in NeurIPS, 2014. 1, 2

[2] J. C. Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. R. Lanckriet, R. Levy, and N. Vasconcelos, “On the role of correlation and abstraction in cross-modal multimedia retrieval,” TPAMI, vol. 36, no. 3, pp. 521– 535, 2014. 1, 3

[3] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, “A new approach to cross-modal multi-media retrieval,” in ACM MM, 2010. 1, 3

[4] H. Park, Y. Yoo, and N. Kwak, “Mc-gan: multi-conditional generative adversarial network for image synthesis,” in BMVC, 2018. 1, 3

[5] Y. Verma and C. V. Jawahar, “Im2text and text2im: Associating images and texts for cross-modal retrieval.” in BMVC, 2014. 1

[6] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in CVPR, 2017. 1, 2, 5

[7] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in ICCV, 2017. 1, 2, 4, 5

[8] H. Tang, H. Liu, and N. Sebe, “Unified generative adversarial networks for controllable image-to-image translation,” IEEE TIP, vol. 29, pp. 8916–8929, 2020. 1

[9] G. Liu, H. Tang, H. Latapie, and Y. Yan, “Exocentric to egocentric image generation via parallel generative adversarial network,” in ICASSP, 2020. 1

[10] Y. Li, Z. Gan, Y. Shen, J. Liu, Y. Cheng, Y. Wu, L. Carin, D. Carlson, and J. Gao, “Storygan: A sequential conditional gan for story visualization,” arXiv:1812.02784, 2018. 1

[11] T. Qiao, J. Zhang, D. Xu, and D. Tao, “Mirrorgan: Learning text-toimage generation by redescription,” in CVPR, 2019. 1, 2

[12] M. Zhao, T. Li, M. Abu Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, “Through-wall human pose estimation using radio signals,” in CVPR, 2018. 1

[13] L. Chen, S. Srivastava, Z. Duan, and C. Xu, “Deep cross-modal audio-visual generation,” in ACM MM Workshop, 2017. 1, 2, 3, 5, 6, 7

[14] W. Hao, Z. Zhang, and H. Guan, “Cmcgan: A uniform framework for cross-modal visual-audio mutual generation,” in AAAI, 2018. 1, 2, 3, 6, 7

[15] A. Rouditchenko, H. Zhao, C. Gan, J. McDermott, and A. Torralba, “Self-supervised audio-visual co-segmentation,” in ICASSP, 2019. 1

[16] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, “Multimodal deep learning,” in ICML, 2011. 1

[17] T.-H. Oh, T. Dekel, C. Kim, I. Mosseri, W. T. Freeman, M. Rubinstein, and W. Matusik, “Speech2face: Learning the face behind a voice,” in CVPR, 2019. 1

[18] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” arXiv:1411.1784, 2014. 1

[19] L. Deng, J. Droppo, and A. Acero, “Enhancement of log mel power spectra of speech using a phase-sensitive model, vol. 12, no. 2, pp. 133–143, 2004. 1

[20] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, “Creating a multi-track classical musical performance dataset for multimodal music analysis,” TMM, pp. 522–535, 2016. 2, 5

[21] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, “Video-to-video synthesis,” in NeurIPS, 2018. 2

[22] H. Tang, D. Xu, Y. Yan, P. H. Torr, and N. Sebe, “Local class-specific and global image-level generative adversarial networks for semantic-guided scene generation,” in CVPR, 2020. 2

[23] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-toimage translation,” in CVPR, 2018. 2, 3, 4

[24] H. Tang “Xinggan for person image generation,” in ECCV, 2020. 2

[25] C. Ledig, L. Theis, F. Husza´r, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang et al., “Photo-realistic single image super-resolution using a generative adversarial network,” in CVPR, 2017. 2

[26] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, “High-resolution image synthesis and semantic manipulation with conditional gans,” in CVPR, 2018. 2

[27] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in ICCV, 2017. 2

[28] X. Chen, C. Xu, X. Yang, L. Song, and D. Tao, “Gated-gan: Adversarial gated networks for multi-collection style transfer,” TIP, vol. 28, no. 2, pp. 546–560, 2019. 2

[29] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, “Multi-channel attention selection gan with cascaded semantic guidance for cross-view image translation,” in CVPR, 2019. 2

[30] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” in ICML, 2019. 2, 3

[31] H. Tang and N.Sebe, “Dual attention gans for semantic image synthesis,” in ACM MM, 2020. 2

[32] W. Shen, “Learning residual images for face attribute manipulation,” in CVPR, 2017. 2, 4
[33] L. Zhao, X. Peng, Y. Tian, M. Kapadia, and D. Metaxas, “Learning to forecast and refine residual motion for image-to-video generation,” in ECCV, 2018. 2, 4
[34] R. K. Davenport, C. M. Rogers, and I. S. Russell, “Cross modal perception in apes,” Neuropsychologia, vol. 11, no. 1, pp. 21–28, 1973. 3
[35] J. Vroomen and B. d. Gelder, “Sound enhances visual perception: cross-modal effects of auditory organization on vision,” Journal of experimental psychology: Human perception and performance, vol. 26, no. 5, p. 1583, 2000. 3
[36] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit, “One model to learn them all,” arXiv:1706.05137, 2017. 3
[37] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, “Taskonomy: Disentangling task transfer learning,” in CVPR, 2018. 3
[38] A. Spurr, J. Song, S. Park, and O. Hilliges, “Cross-modal deep variational hand pose estimation,” in CVPR, 2018. 3
[39] C. Hu, D. Li, Y.-Z. Song, and T. M. Hospedales, “Now you see me: Deep face hallucination for unviewed sketches,” in BMVC, 2016. 3
[40] A. Duarte, F. Roldan, M. Tubau, J. Escur, S. Pascual, A. Salvador, E. Mohedano, K. McGuinness, J. Torres, and X. G. i Nieto, “Wav2pix: Speech-conditioned face generation using generative adversarial networks,” in ICASSP, 2019. 3
[41] C.-H. Wan, S.-P. Chuang, and H.-Y. Lee, “Towards audio to scene image synthesis using generative adversarial network,” in ICASSP, 2019. 3
[42] H. Tang, D. Xu, W. Wang, Y. Yan, and N. Sebe, “Dual generator generative adversarial networks for multi-domain image-to-image translation,” in ACCV, 2018. 3
[43] N. Khan, B.-W. Hong, A. Yezzi, and G. Sundaramoorthi, “Coarse-to-fine segmentation with shape-tailored continuum scale spaces,” in CVPR, 2017. 3
[44] M. Pedersoli, A. Vedaldi, J. Gonzalez, and X. Roca, “A coarse-to-fine approach for fast deformable object detection,” PR, vol. 48, no. 5, pp. 1844–1853, 2015. 3
[45] D. Cristinacce, T. F. Cootes, and I. M. Scott, “A multi-stage approach to facial feature detection.” in BMVC, 2004. 3
[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in CVPR, 2009. 4
[47] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in MICCAI, 2015. 5
[48] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in CVPR, 2016. 5
[49] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in ICLR, 2015. 5
[50] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in NeurIPS, 2017. 6
[51] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in NeurIPS, 2016. translation task.

APPENDIX

A. SOURCE CODE

train.py file:

```python
import time
from options.train_options import TrainOptions
from data import create_dataset
from models import create_model
from util.visualizer import Visualizer

if __name__ == '__main__':
    opt = TrainOptions().parse()
    dataset = create_dataset(opt)
    dataset_size = len(dataset)
    print('The number of training images = %d' % dataset_size)

    model = create_model(opt)  # create a model given opt.model and other options
    model.setup(opt)  # regular setup: load and print networks; create schedulers
    visualizer = Visualizer(opt)  # create a visualizer that display/save images and plots
    total_iters = 0  # the total number of training iterations
    for epoch in range(opt.epoch_count, opt.niter + opt.niter_decay + 1):    # outer loop for different epochs; we save the model b
        epoch_start_time = time.time()  # timer for entire epoch
        iter_data_time = time.time()    # timer for data loading per iteration
        epoch_iter = 0                  # the number of training iterations in current epoch, reset to 0 every epoch

        for i, data in enumerate(dataset):  # inner loop within one epoch
            iter_start_time = time.time()  # timer for computation per iteration
            if total_iters % opt.print_freq == 0:
                t_data = iter_start_time - iter_data_time
            visualizer.reset()
            total_iters += opt.batch_size
            epoch_iter += opt.batch_size
            model.set_input(data)         # unpack data from dataset and apply preprocessing
            model.optimize_parameters()   # calculate loss functions, get gradients, update network weights

            if total_iters % opt.display_freq == 0:   # display images on visdom and save images to a HTML file
                save_result = total_iters % opt.update_html_freq == 0
                model.compute_visuals()
                visualizer.display_current_results(model.get_current_visuals(), epoch, save_result)

            if total_iters % opt.print_freq == 0:    # print training losses and save logging information to the disk
                losses = model.get_current_losses()
                t_comp = (time.time() - iter_start_time) / opt.batch_size
                visualizer.print_current_losses(epoch, epoch_iter, losses, t_comp, t_data)
                if opt.display_id > 0:
                    visualizer.plot_current_losses(epoch, float(epoch_iter) / dataset_size, losses)

            if total_iters % opt.save_latest_freq == 0:   # cache our latest model every <save_latest_freq> iterations
                print('saving the latest model (epoch %d, total_iters %d)' % (epoch, total_iters))
                save_suffix = 'iter_%d' % total_iters if opt.save_by_iter else 'latest'
                model.save_networks(save_suffix)

            iter_data_time = time.time()
        if epoch % opt.save_epoch_freq == 0:              # cache our model every <save_epoch_freq> epochs
            print('saving the model at the end of epoch %d, iters %d' % (epoch, total_iters))
            model.save_networks('latest')
            model.save_networks(epoch)

        print('End of epoch %d / %d \t Time Taken: %d sec' % (epoch, opt.niter + opt.niter_decay, time.time() - epoch_start_time))
        model.update_learning_rate()                     # update learning rates at the end of every epoch.
```

test.py file:

```
import os
from options.test_options import TestOptions
from data import create_dataset
from models import create_model
from util.visualizer import save_images
from util import html

if __name__ == '__main__':
    opt = TestOptions().parse()  # get test options
    # hard-code some parameters for test
    opt.num_threads = 0  # test code only supports num_threads = 1
    opt.batch_size = 1  # test code only supports batch_size = 1
    opt.serial_batches = True  # disable data shuffling; comment this line if results on randomly chosen images are needed.
    opt.no_flip = True  # no flip; comment this line if results on flipped images are needed.
    opt.display_id = -1  # no visdom display; the test code saves the results to a HTML file.
    dataset = create_dataset(opt)  # create a dataset given opt.dataset_mode and other options
    dataset_size = len(dataset)
    opt.num_test = 300
    print('The number of testing audio = %d' % dataset_size)
    model = create_model(opt)  # create a model given opt.model and other options
    model.setup(opt)  # regular setup: load and print networks; create schedulers
    # create a website
    web_dir = os.path.join(opt.results_dir, opt.name, '%s_%s' % (opt.phase, opt.epoch))  # define the website directory
    webpage = html.HTML(web_dir, 'Experiment = %s, Phase = %s, Epoch = %s' % (opt.name, opt.phase, opt.epoch))
    # test with eval mode. This only affects layers like batchnorm and dropout.
    # For [pix2pix]: we use batchnorm and dropout in the original pix2pix. You can experiment it with and without eval() mode.
    # For [CycleGAN]: It should not affect CycleGAN as CycleGAN uses instancenorm without dropout.
    if opt.eval:
        model.eval()
    for i, data in enumerate(dataset):
        #if i >= opt.num_test:  # only apply our model to opt.num_test images.
        #    break
        model.set_input(data)  # unpack data from data loader
        model.test()  # run inference
        visuals = model.get_current_visuals()  # get image results
        audio_path = model.get_audio_paths()  # get audio paths
        if i % 5 == 0:  # save images to an HTML file
            print('processing (%04d)-th image... %s' % (i, audio_path))
        save_images(webpage, visuals, audio_path, aspect_ratio=opt.aspect_ratio, width=opt.display_winsize)
    webpage.save()  # save the HTML
```

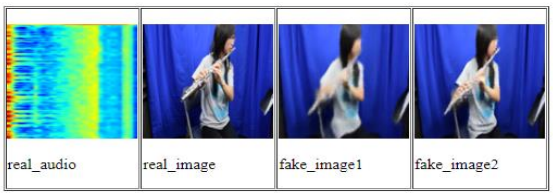
B. SCREENSHOTS

bassoon01_2400

cello01_1400

clarinet01_2400

flute01_1400



flute01_2400

oboe04_1400

oboe04_2400

trombone02_2400