StyleWav: Guiding Image Synthesis Using Audio

Jerry Ngo

700 College St, Beloit, WI 53511

ngoph@beloit.edu

Abstract. Recent works in multimodality systems have enabled image generations with instruction from another domain. Specifically, Contrastive Language-Image Pre-training (CLIP) allows multiple downstream tasks such as image synthesis, 3D object generation, and style transfer using text. However, the audio-image generation has not been much explored. In this work, we introduce StyleWav, an algorithm that is adapted from StyleCLIP. Instead of using text guidance as in StyleCLIP, StyleWav uses audio representation from Wav2CLIP to guide the generated image using StyleGAN. We use synthesized images as a concrete benchmark for the transferred knowledge from the distillation process from CLIP to the audio domain. For further application, the model's code is public at: https://github.com/Jerry2001/StyleWav.

1 Introduction

Traditionally, machine learning models have been trained using the supervised learning paradigm, where the training dataset has been carefully labeled by humans. However, the recent introduction of self-supervised learning has attracted a lot of attention because they only require unlabelled data for training [16]. Unlike supervised learning, in self-supervised learning, the model updates its weights through self-supervision signals from a large amount of unlabelled data. One popular approach to self-supervised learning is to use contrastive loss, which minimizes the representation distance between samples from the same class and maximizes the distance

between samples from different classes. [8] The technique has shown impressive results in different domains like vision, natural language processing and speech [16]. In this work, we would leverage the utility of one supervised learning model called Contrastive Language-Image Pretraining (CLIP) [14].

CLIP is a self-supervised vision-language model that is trained on 400 million (image, text) pairs collected from various websites [14]. After pretraining, the model could be used with a wide range of downstream tasks without the need to be finetuned (zero-shot transfer). The main functionality of CLIP is to provide the similarity score between a text prompt and an image. And this could be leveraged to be used as a loss function of vision and language algorithm because of its zero-shot transfer property.

Since CLIP is a big model that has learned a lot of vision and language concepts on the world already, it is possible to extend its capability to another modality. Wav2CLIP [16] is a distilled model from CLIP that could process three modalities: vision, language, and audio. In this model, the audio representation is contrastively trained with the image. Thus, Wav2CLIP maps the audio representation to the join vision-language representation space. The audio is also corresponding to the image so that we can calculate the similarity rating between them and perform zero-shot transfer on downstream tasks and one of them is image generation[16].

Image generation or synthesis is a task in machine learning where you want to generate a fake image that usually follows a specific requirement or distribution. And one of the popular frameworks to synthesize images is generative adversarial networks (GAN) [4]. Wav2CLIP illustrates its ability to guide image-generation with audio by using a GAN model called VQGAN [3], [16]. Even though the generated image shows some semantics detail, it isn't aesthetically close to the human's view of the concept. In this work, we attempt to synthesize more realistic images with audio by switching the generator to StyleGAN2 [12] and narrowing down the domain to human faces. The technique we use here is latent space optimization while using CLIP as the loss function.

2 Model

In this section, I present the component of the StyleWav components. On the higher-level idea, StyleGAN is used to generate the image. The

synthesized image is then fed into CLIP and mapped into a joint representation space. At the same time, Wav2CLIP map a 5-second audio file to the same space. Since now both the audio and image are embedded in the same latent space, we can compute the cosine similarity between them to get the similarity score. The detail is described in the following figure.

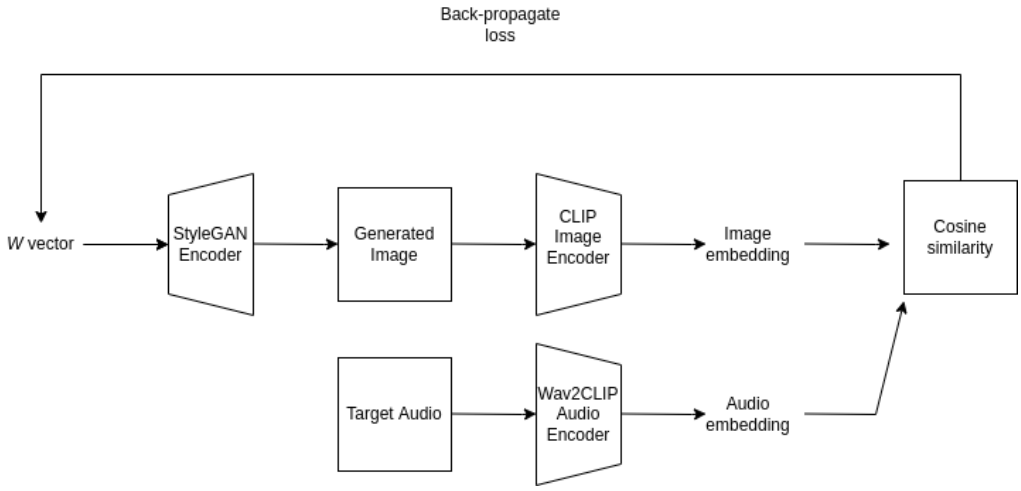

Figure 1: We first initialize the W vector with a mean vector of the StyleGAN2 representation space. Such vector is fed into the StyleGAN2 encoder to generate an image. After getting the embedding of the image from CLIP and the audio embedding from Wav2CLIP, we calculate the cosine similarity between them. This will tell us how far the image is from the audio right now. We then back-propagate the cosine similarity to update the W vector, and the process repeat until we reach the set number of steps.

3 CLIP

CLIP includes 2 two networks: the text head and image head. And they are jointly trained with natural language supervision. For the vision head, there are two different options of architecture ResNet [14], [7] and Vision Transformer [2], and each architecture has different models with 5 ResNets and 3 Vision Transformers [14]. For the text head, they use a modified version of the Transformer [15]. For the 12-layer version of it, they use a

vocabulary size of 49408 to represent the input text. And, the maximum sequence size is 76 because of the computational constraint.

To perform zero-shot image-text prediction, we choose a set of N target label texts and then get the list of their embeddings from the text head. We then choose a batch of M images we want to match with the label and get their embeddings from the image head. After having all the embeddings, we can construct a Sim similarity matrix with the size of NxM by finding the cosine similarity distance between every pair of image and text. The similarity between the text prompt index i and the image prompt index j is located at $Sim_{(i,\ j)}$.

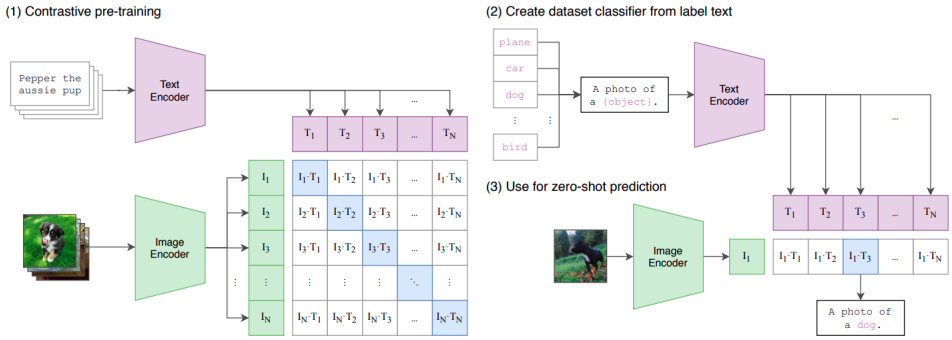

Figure 2: In the first phase (1), two heads are contrastively jointly-pretrained where we want to maximize the diagonal of the similarity matrix (the score between the image and the matching prompt). After pretraining, we can perform a zero-shot prediction. In the right images, we have a photo of a dog and a list of label we want to match it with. Notice that we won't use one-word labels but instead put each label in a prompt with the form of "A photo of label" because such a prompt fall within the language distribution of the text head, thus, yield a better result. [14]

4 Wav2CLIP

4.1 Knowledge Distillation

Knowledge Distillation is the process where we transfer knowledge from a ensemble or a bigger model into a smaller and distilled model [8]. The

simplest way to distill knowledge is "by training it on a transfer set and using a soft target distribution for each case in the transfer set that is produced by using the cumbersome model with a high temperature in its softmax.[8]"

When first introduced, knowledge distillation was mostly utilized to compress the model size so that we could deploy them in production. However, it is also possible to use this technique to transfer knowledge between domains. The paper [5] has shown that it is possible to transfer the knowledge from a model that learned the representation of labeled data in one modality to unlabeled data in another modality using paired images from both modalities. The same concept has been applied to CLIP to obtained a distilled version of CLIP in the audio domain.

4.2 Dataset

As mentioned earlier, to transfer knowledge from one modality to another, we need a dataset with pair of data in learned modality and new modality. Since CLIP has two modalities, text and image, we need to find a dataset that either has text-audio or image-audio. The dataset they used was VGG-Sound [16], [1].

VGG-Sound is a large-scale Youtube Audio-Visual dataset where each record is a short clip of Youtube video and the corresponding classes. There are more than 200000 videos and 310 classes [1]. Note that this distribution is much smaller than CLIP's.

4.3 Wav2CLIP Distillation

In the distillation process, the weight of vanilla CLIP's image head is frozen [16]. They then put a multilayer perceptron on top of bother embeddings as a projection layer. Following the original CLIP paper, they also used a contrastive loss to distill the audio encoder. The loss function between two embedding is:

$$CXLOSS = L(f(Image), Audio) + L(Image, g(Audio))$$

where f, g are projection functions and L: contrastive loss [16]

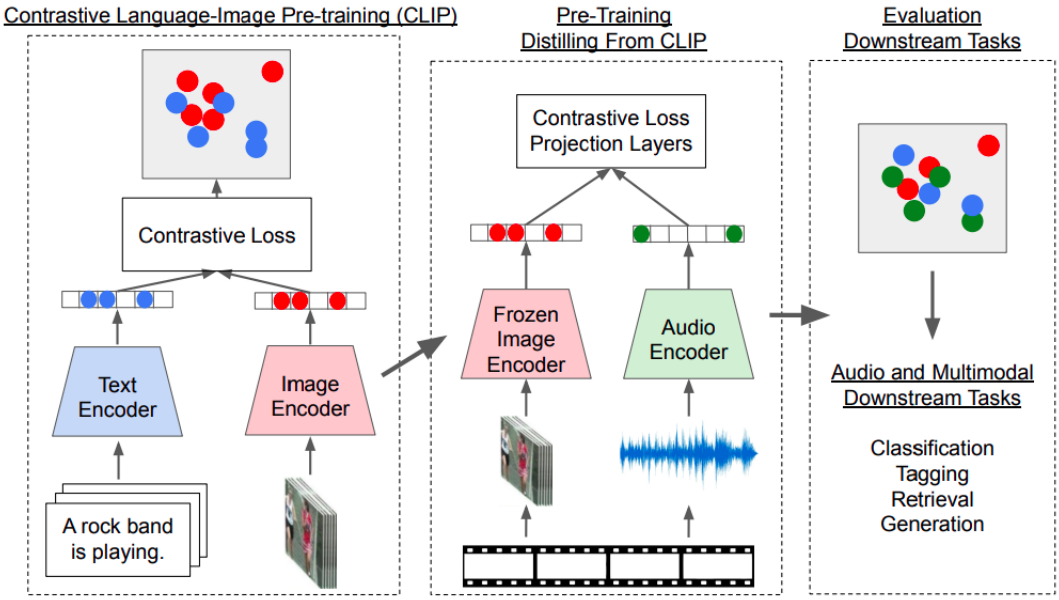

Figure 3: In the far right image, after the audio encoder has been pretrained, the audio is embedded in the same representation space as text and image. Thus, we could perform multiple downstream task with zero-shot transfer. [16]

5 Generative Adversarial Network

Generative Adversarial Network (GAN) is a class of generative models. The model comprises two components: the discriminator and the generator [4]. In the higher-level idea, the generator is used to generate an image, whereas the discriminator is meant to be able to tell a fake image from a real image.

GAN procedure begins with the generator synthesizing an image from a random vector. These images will then be mixed with real ground-truth images, and the discriminator will generate a probability vector to decide which image is fake and which one is real. If the generator is able to tell the fake image from the real image, the generator weight will be updated. Otherwise, the discriminator weight will be updated. This is similar to the cop and counterfeiter analogy. Similar to the generator, the counterfeiter

tries to fake a coin and if he gets caught, he will try to get better at making counterfeit coins next time. Since in each phase, either the generator or the discriminator weight will be updated, it explains the "adversarial" in GAN's name. However, before 2018, most of the work on GAN focused on

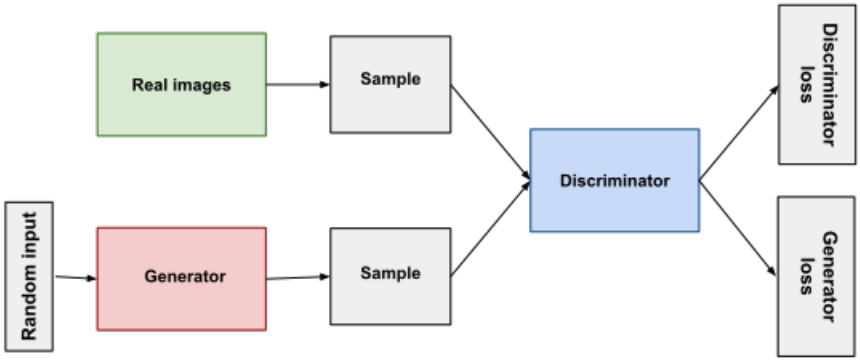

Figure 4: GAN Architecture (from https://developers.google.com/machine-learning/gan/gan_structure)

the generator component of the model. So, even though the quality and the resolution of the image were growing rapidly, the generate still remained a black box [11]. The StyleGAN architecture was proposed to solve such issues.

5.1 StyleGAN

In the traditional generator, the generator will take a random vector z from the Z latent space and synthesize the image. And this has been hypothesized to cause some degree of unavoidable entanglement. So to solve the issue, StyleGAN maps the z vector through an 8-layer MLP into a vector $w \in W$. We have the transformation function of $f : Z \to W$. The dimensionality of W and Z space is both 512 [11].

"Learned affine transformations then specialize w to styles $y = (y_s, y_b)$ that control adaptive instance normalization (AdaIN) opertions after each convolution layer of synthesis network g." We have the AdaIN operation

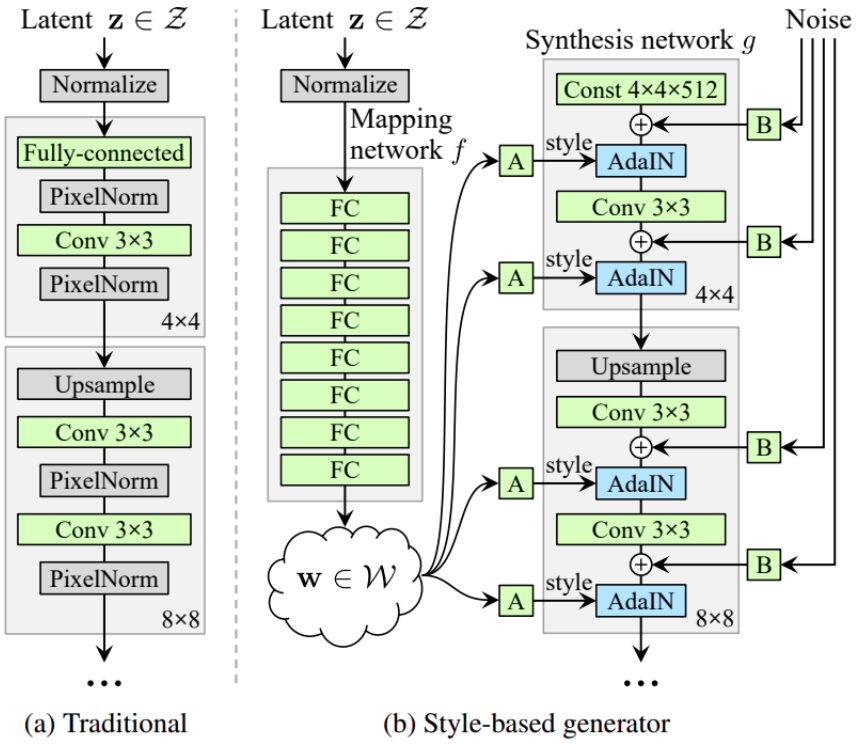

Figure 5: StyleGAN Architecture [11]

formula to be:

$$\text{AdaIN}(x_i, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\delta(x_i)} + y_{b,i}. \text{ [11]}$$

Besides styles vector, Gaussian noise is added after each convolution layer. The effect of noise input has enabled the model to generate stochastic aspects of human, such as the placement of freckles and hairs [11].

The combination of all changes has enabled StyleGAN to generate high-quality, realistic images (Figure 6). The latent space of StyleGAN is also disentangled which allows us to have fine-grained control over which style we want a photo to have [17].

Figure 6: StyleGAN Example: The set of generated images from StyleGAN model trained on FFHQ dataset (config F) [11]

5.2 StyleGAN2

In this paper, we will use an improved version of StyleGAN called StyleGAN2 [12], which was introduced in 2020. We won't go into a detailed discussion of the model. However, in StyleGAN2, the model tries to eliminate artifacts such as droplet in generated photos. There are also big architecture changes such as moving noise addition and including a demodulation operation [12].

6 Current State of Audio to Image Generation using Wav2CLIP

As the audio embedding corresponds to the image embedding in the Wav2CLIP space, the author demonstrates how audio could guide image generation. The architecture they used is VQGAN, and it was trained on the ImageNet dataset, which includes 14197122 images with 80000 nouns. Thus this model has the capacity to generate a vast amount of concepts. Figure. 7 shows their result.

Figure 7: Text/Audio to Image Generation with VQGAN-CLIP [16]

As we can see from the second row of Figure. 7, the generated images have some semantic information from the audio. For example, the street music audio has a piano, other instruments and a man in the generated image. However, those generated images are not aesthetically beautiful and are very low-resolution. However, the result using text guidance from the top row is very similar to the audio. It could be because the used model covers a large scope of concepts. Thus, in this work, we want to change the choice of model to StyleGAN2, which is more dedicated to one concept per model and also has more disentangled latent space.

7 Experiments

In this experiment, we use StyleGAN2 trained on Flickr-Faces-HQ or FFHQ dataset with high-resolution images of the face. We then attempt to

generate human images from using human audio from the Common Voice dataset. The human voice has been researched to reveal the speakers age, sex, education, and origin [10]. Thus, we are going to evaluate how close the generated image could get to the speaker trait giving only the audio of the speaker.

7.1 Dataset

Common Voice is a public dataset maintained by Mozilla. The dataset has been voluntarily contributed by a diverse set of people, including non-English speakers, people of color, and disabled people...Each audio is also accompanied by metadata that includes the age, gender, the accents, and locale of the speaker []. Since Wav2CLIP is trained on 5-second audio and Common Voice has a different length for each recording, we will only select audio that falls within 5-5.5 seconds of length and if the sentence they speak has more than 6 words in it.

7.2 Sex

In this experiment, we want to see if the generated images reflect the speaker's sex. We randomly sampled 100 female audio and 100 male audio and then generated images from them using the following parameter.

- Type of generation (experiment_type): free generation
- The number of optimization step (number_of_optimization_step): 10
- Do we use a value for the seed instead of randomize it (use_seed): true
- Value of the seed (seed): 0

7.2.1 Generated Images

We have figure 8 with generated images from female audio and figure 9 with generated images from male audio. There are obvious facial differences between two sets of population like hair and facial structure. We could

then quantitatively evaluate generated images to see whether one set of population represents one specific sex more than the other using CLIP. Figure 8.

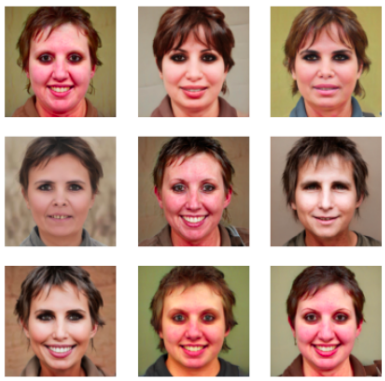

Figure 8: Female audio

Figure 9: Male audio

7.2.2 CLIP Score and T-Test

For each image in each population, we want to calculate the similarity between it and two prompts: "this is a photo of a boy" and "this is a photo of a girl." Because we now have two similarity vectors between one set of generated images and two prompts representing two sexes, we can perform a Welch T-test to see if the two vectors have significantly two different means. The result is in the table below:

	Male photos	Female photos
this is a photo of a boy	0.26	0.241
this is a photo of a girl	0.252	0.256
have significantly two different means	true	true

7.2.3 Discussion

We can see that generated photos using male audio are ranked close to the prompt "this is a photo of a boy" compared to the prompt "this is a photo of a girl," and the other way applies for generated photos using female

audio. Since the mean of the similarity score between the two populations are significantly different, we conclude that Wav2CLIP learned the association between the speaker's voice and their sex. Even though the generated images with audio guidance aren't all realistic as non-guiding StyleGAN2 images, they have fewer artifacts and they are also more consistent and less abstract than the result using VQGAN with Wav2CLIP.

8 Limitation

Besides sex, we also tried out different traits of the speaker, such as locale and age, but the results were relatively random. We hypothesize this is due to the mismatch in distribution between VGGSound dataset and CLIP training data. More than that, in VGGSound, attributes such as age and locale are not equally distributed and skew towards young people and people from English-speaking countries.

9 Conclusion

In this work, we introduced the StyleWav algorithm that could guide human generation using audio. We also show that the generated image's features correspond to the sex of the speaker. However, the model still couldn't capture different traits such as age and locale, which could be traced back to this mismatch in dataset distribution.

References

[1] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. 4 2020.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. 10 2020.

[3] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. 12 2020.

[4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. 6 2014.

[5] Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. 7 2015.

[6] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. 6 2021.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. 3 2015.

[9] Weiran Huang, Mingyang Yi, and Xuyang Zhao. Towards the generalization of contrastive self-supervised learning. 11 2021.

[10] Swati Johar. Emotion, affect and personality in speechthe bias of language and paralanguage. 2016.

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. 12 2018.

[12] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. 12 2019.

[13] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. 3 2021.

[14] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. 2 2021.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 6 2017.

[16] Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2clip: Learning robust audio representations from clip. 10 2021.

[17] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. 11 2020.