SCORE! Technical documentation and project overview

This document forms both the final report of SCORE! and the documentation for deliverable 2 of the project.

This report briefly describes the technical work done for SCORE! (NWO kiem project 314-98-121), the main conclusions from the project and its deliverables. For a more in-depth overview of the current status, conclusions and future directions, see the third deliverable report.

We present the following technical deliverables:

- A **demonstrator script**, available at https://github.com/pbloem/score
- A library of **visual models**, available at https://github.com/pbloem/pixel-models
- A library of **sequence models**, available at https://github.com/pbloem/language-models

Project overview

The SCORE! project aims to ascertain to what extent modern deep learning methods allow us to generate audio for a given video in an *unsupervised* way. That is, without relying on human annotation, or existing mappings between video/audio material.

Names of scholars involved
Dr. Peter Bloem, VU University
Dr. Victor de Boer, VU University

Names of the partners involved
Gregory Markus, The National Institute Sound and Vision
Dara Smith, Lakker

Actual and expected results
Our primary deliverable, stated in the research proposal, was a technical demonstrator to be informally evaluated by audiovisual artists. We performed such an evaluation with the artist Dara Smith of the duo Lakker.

A more elaborate evaluation, using software integrated into professional audio-production software, and a larger sample of audiovisual professionals is underway as part of a VU Master's project, expected to finish June 2019.

We also developed several secondary software deliverables not described in the project proposal. These are detailed below in the body of the report.

Changes and developments with respect to the original plan
Apart from certain technical changes to the model described in the project proposal, there were no major deviations to the project plan. These technical changes are detailed below in the body of the report.

Evaluation of the project / Future prospects of the project (e.g. prolonged co-operation with the partner, new research proposals)

For a detailed evaluation see the third project deliverable. Two student projects are currently underway to investigate additional aspects of the project. Based on the results of these, we will decide how to continue with the project, and whether to collaborate on future project proposals. These projects are expected to finish June 2019. Regardless of external funding, the collaboration between Sound and Vision will continue in the coming years.

The SCORE project was presented at the Instrumental Shifts Symposium of the 2019 RE:WIRE festival in The Hague.

How did you find your partner(s)?
The Institute for sound and vision was approached based on previous collaborations with Dr. de Boer. Dara Smith had previously collaborated with Gregory Markus.

To which extent did either the Chamber of Commerce or the CLICK NL networks help in finding the partner(s)? In what respect did you benefit from the TKI CLICK NL for the Creative Industries when setting up your project or during the realisation of your project?

We did not use these resources to find our partners or in the course of the project.

Primary deliverable: demonstrator script

The original plan for SCORE! called for two variational autoencoders with a mapping from one latent space to another. See figures 1 and 2 for an explanation.

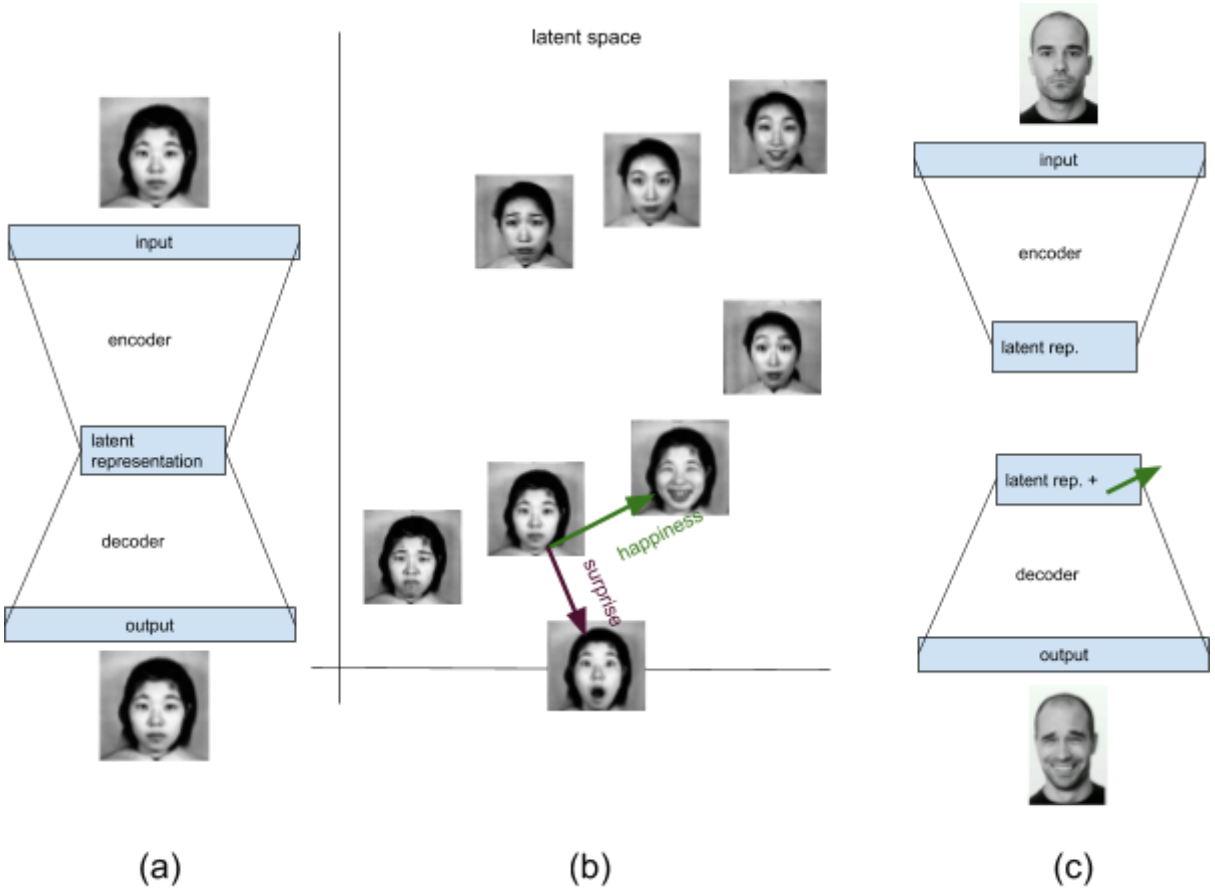

Figure 1: A schematic illustration of the principle of latent representations. **(a)** A (variational) autoencoder maps the input to a low-dimensional latent representation. Its training objective is to reconstruct the input from the latent representation. **(b)** The training data mapped to latent space. The data is clustered by person, but directions in the latent space correspond to important, high-level variations in the data (in this case facial expressions). **(c)** We can take a new image of a person with a neutral expression, map it to a latent representation (a point in latent space), move it in the "happiness" direction, and decode it to make the person smile.

To provide an early testing prototype, we opted for a demonstrator built entirely from pretrained models. For the auditory component, the Magenta MusicVAE[1] provided a useful pretrained MIDI autoencoder.

For the visual component, no pretrained autoencoders were available that provide solid latent features for the general visual domain. Since we only required the encoder component, we opted instead to use (part of) a pre-trained classifier network. These networks have been shown previously to result in highly descriptive feature-vectors, that allow many tasks, including visual art.[2]

Since the music decoder expects inputs in to be distributed as a standard normal distribution, and this is not guaranteed for the features from the image classifiers, we apply principal component analysis to whiten the feature and reduce the dimensionality to the required degree.

[1] https://magenta.tensorflow.org/music-vae
[2] See for instance https://www.youtube.com/watch?v=C6v2Z0IAcY4

Our script allows for the use of both the Inception v3 network [1], and the slightly smaller and faster MobileNet v2 [2].

This resulted in a command-line script that can generate MIDI music for any given video, with special facilities for using the sound and vision archival video.

The script is hosted in the main SCORE! repository, here:
https://github.com/pbloem/score
with documentation provided in the file README.md.

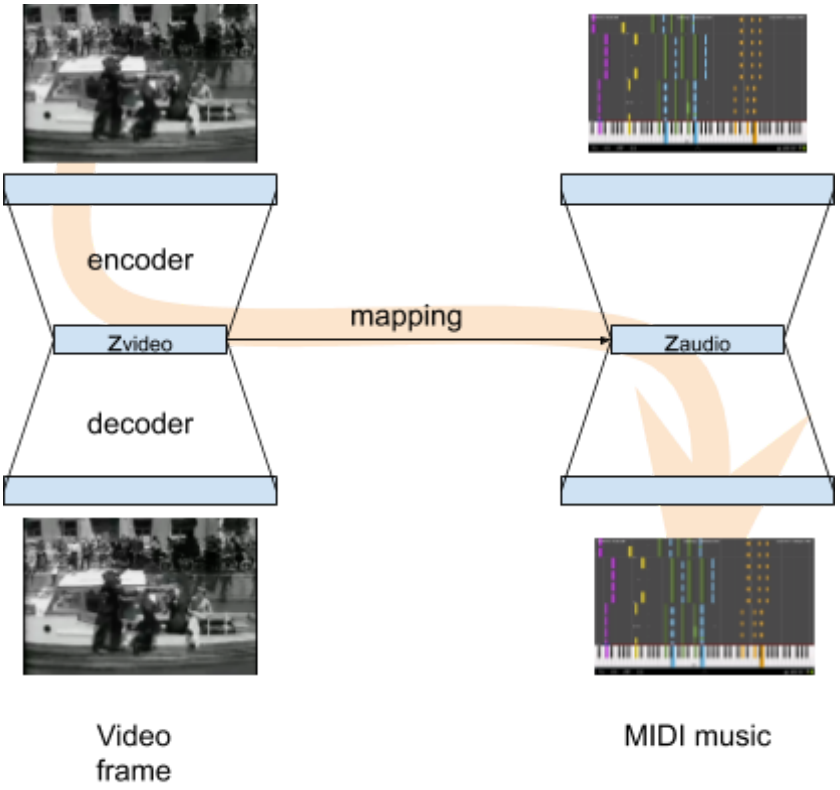

Figure 2: A schematic representation of the original proposed system. Two auto-encoders are trained independently on frames of video and on MIDI music. We establish a mapping between the two latent spaces. We then generate music for a video by encoding the video frame to a latent space, mapping this to the audio latent space, and decoding into music. In practice, we ended up using a pretrained classifier as the image encoder, and whitening the features using PCA.

The repository also contains several code examples for loading, processing and training autoencoders on the sound and vision archive.

Results

Some example output from the demonstrator, using video from the sound and vision archive can be found through the SCORE! Website:
https://pbloem.github.io/score

This demonstrator was sufficient to answer the main question behind the SCORE! Project: whether such automatically generated were sufficient to provide a audiovisual artist with a working basis, or a spark of inspiration. The results of these informal tests can be found in the third deliverable report. A more elaborate evaluation is underway as part of a master's thesis to be concluded towards the end of 2019.

Secondary deliverables: training autoencoders on sound and vision data

The second objective of the SCORE! project was to lay the groundwork for a larger project: to investigate and study the existing research for possible extensions of the project.

Custom models in the visual domain

One option for extending the model is to train autoencoders that are specific to the visual dataset: that is, to train variational autoencoders specifically on the frames of the sound and vision archive. This would result in a model with latent dimensions that are fitted specificall to the data.

Figure 3 shows the result of an early investigation. While similar concepts seem to be being grouped together, an informal investigation of the model and its reconstructions suggested that the model was focusing much more on representing low-level visual features (such as color) than on high level semantic features (such as the object contained in the image).

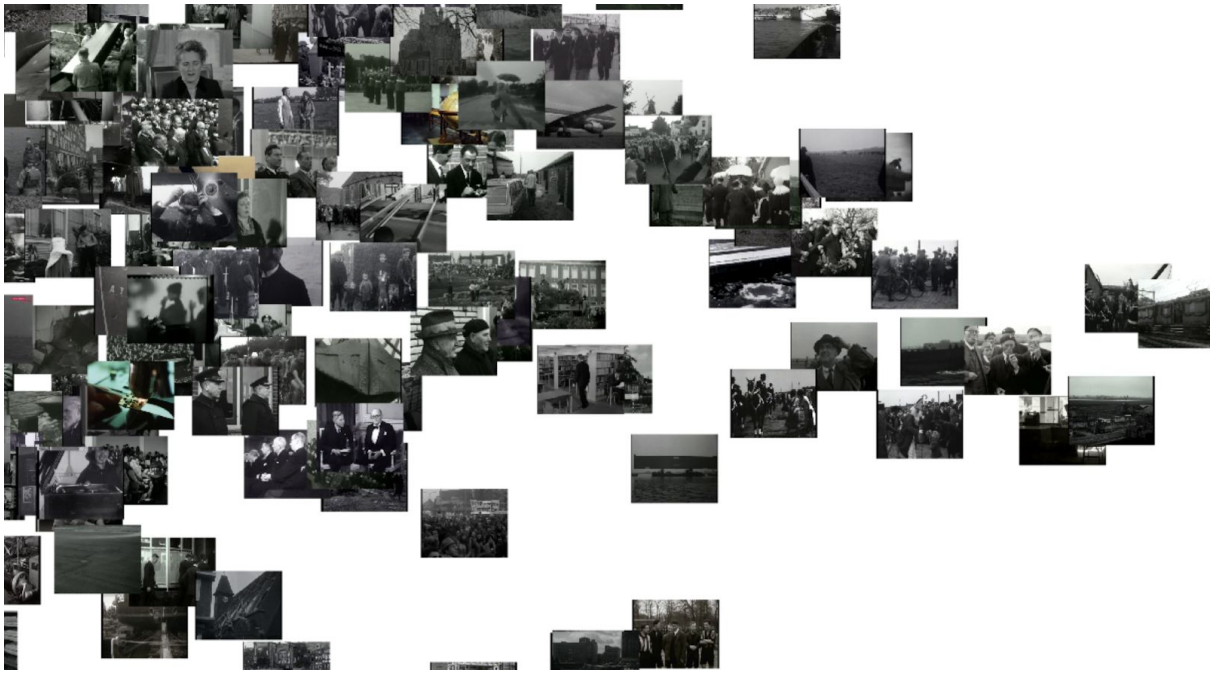

Figure 3 A close-up of part of a 2D latent space from an autoencoder trained on the frames of the sound and vision video archive. Note that in higher dimensional spaces the images can be laid out more naturally.

The takeaway is that such models are likely less suited to the task than the classifier models used in the demonstrator script. The main difference is that the latter, while trained on data

from a different domain, are trained on *labeled images*, allowing them to more directly model high-level semantic concepts.

One way to make an unsupervised model more attentive to the content of the image, is to let the decoder model the low-level visual attributes autoregressively. This allows the latent space to focus more strongly on the semantic content of the image.

We investigated this option by creating implementations of two strongly related models the PixelCNN [3] and the PixelVAE [4] models. Our implementations are available at https://github.com/pbloem/pixel-models.

Since no well-tested implementations of these models in pytorch were previously available, this library has already been adopted by the community for use in other projects.

A full investigation of whether such models can be used to lift unsupervised training to the level of the supervised transfer learning used in the demonstrator is outside the scope of the project, but provides a strong first research objective for a follow-up project.

Custom models in the auditory domain

While the MusicVAE model provides reasonable music-generation from a latent space, the main takeaway form the results of the demonstrator script is that for an audio/video combination to feel natural, there needs to be an *immediate* response. That is, when something happens in the video, like a scene change or a sudden visual event, the music should immediately change in response.

With the MusicVAE model, this is not possible, since is generates music in 2 or 16 second chunks. That means that any response to the video feed will always be gradual. This leads to another research question: can we build an encoder model that reconstructs music from a high-frequency sequence of latent vectors, in such a way that manipulating the latent vectors leads to natural changes in the music?

If so, we could generate a sequence of latent visual vectors from the video frames, and add it to the latent music vectors (or feed it to the decoder as is) to generate music that responds to the video immediately (while retaining a coherent structure over time).

Again, designing and building such a custom model is outside the scope of the project, but as with the visual models, we implemented some candidate models as a way of investgating the landscape of available approaches. These include the often-used autoregressive RNN and the variational autoencoder of Bowman et al. [5].

These models are available at
https://github.com/pbloem/language-models

Because no good implementations of the Bowman model were available in Keras, this library too, has been adopted by the community for use in other projects.

Our implementations focus on language rather than music, to simplify the problem and allow easier validation. A student project is currently underway to test some of these approach in the musical domain.

References

[1] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
[2] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
[3] Van den Oord, Aaron, et al. "Conditional image generation with pixelcnn decoders." *Advances in neural information processing systems*. 2016.
[4] Gulrajani, Ishaan, et al. "Pixelvae: A latent variable model for natural images." *arXiv preprint arXiv:1611.05013* (2016).
[5] Bowman, Samuel R., et al. "Generating sentences from a continuous space." *arXiv preprint arXiv:1511.06349* (2015).