University of Oslo

IN5310 - Advanced Deep Learning for Image Analysis

Group 2 - Berezin, Ilya and Kumar, Nikhil

August 26, 2024

Project 2: Conditional Image Generation

Reproducibility

The code used for this report can be accessed at:
https://github.uio.no/2023-s2-in5310-in9310/g02-p2

1 Abstract

Image reconstruction is a fundamental task in computer vision and image processing, with applications spanning from medical imaging and remote sensing to image super-resolution and artistic style transfer. In recent years, deep learning models have emerged as a powerful approach for tackling image reconstruction challenges, offering remarkable improvements in accuracy and quality over traditional methods [4]. These deep learning models are especially effective when they are provided with conditioned inputs, enabling them to generate highly detailed and contextually relevant reconstructions [3].

Conditioned image reconstruction refers to the process of generating an image from a set of input conditions or constraints that guide the reconstruction process. By leveraging these conditioned inputs, deep learning models can produce reconstructions that are not only visually appealing but also aligned with specific requirements.

This report explores the fascinating realm of deep learning (DL) models for image reconstruction with conditioned inputs. It delves into the underlying concepts, methodologies, and the state-of-the-art approaches that have revolutionized image reconstruction tasks. The utilization of conditioned inputs adds an extra layer of control and customization to the reconstruction process, making it a versatile tool in a wide range of applications.

The DL models used for the reconstruction is Conditioned Variational Auto-encoder (CVAE) [2]. The CVAE model encapsulates the capability to capture complex, non-linear relationships [1]. This facilitates a nuanced understanding of the preferences, enabling more accurate image reconstructions.

In this project, our primary objective is to generate 10 images which are jointly optimal for pleasing Moira and Ferdinando, the second and fourth respondents in the focus group respectively.

2 Methods

2.1 Dataset

For this project, we used the data which consists of 6000 RGB images (with some grayscale instances that we further process to RGB) of various cars with resolution 96px $\times$ 128px, alongside annotations from six members of a focus group. The focus group rated each car on a scale of one to ten, encoded as $si \in \{0, \ldots, 9\}$ for each of the i = 1,...,6 responders. The responders are named Raof, Moira, Louie, Ferdinando, Gragar, and Esther, and their scores are presented in this order.

2.2 Data Processing and Score Extraction

We started our process by setting a consistent seed to ensure the reproducibility of our experiments. This guarantees consistent outcomes across multiple runs, which is essential for a fair comparison of different models or techniques.

To preprocess the data, a series of transformations were applied:

- Images were converted to tensors with 3 channels, ensuring consistent dimensions suitable for neural networks, as some of the instances were represented in grayscale.
- We resized the images to a standard size of 224x224 pixels, an input size for pretrained ResNet18 architecture.
- Reviewer scores, which are our conditions for image generation, were converted into PyTorch tensors.

For the dataloader, we adopted two unique weighting strategies:

- **Product of Scores**: This strategy prioritized images based on the product of scores from both reviewers. It highlights images where both reviewers had mutual agreement on the image quality, by multiplying them.
- **Consensus-Based**: This approach gave prominence to images where there was a consensus between the two reviewers, emphasizing images with close or similar scores.

2.3 Model Description: CVAE_ResNet18_DualEmbedding

The `CVAE_ResNet18_DualEmbedding` is a class that represents a Conditional Variational Autoencoder (CVAE) that integrates the ResNet18 architecture and can handle dual embeddings. Here's its structure and purpose:

1. **Objective**: The model's goal is to generate images based on certain conditions. In our case, these conditions represent preferences of two respondents, Moira and Ferdinando.
2. **Architecture**:
    - **ResNet18 Backbone**: The model leverages the ResNet18 architecture to extract features from the input images. ResNet18 is a deep residual network that has proven its efficiency in image recognition tasks.
    - **Encoder**: After extracting the features using ResNet18, the model concatenates these features with the condition vectors of the two respondents. These concatenated vectors then pass through linear layers (`fc_mu` and `fc_logvar`) to produce the mean and variance of the latent space.
    - **Reparameterization**: This step allows the model to sample from the latent space. By doing this, the model can generate different outputs for the same input, introducing variability in the generated images.
    - **Decoder**: The decoder reconstructs the original image from the latent space. Here, the model concatenates the sampled latent vector with the condition vectors and then decodes this combined vector through a series of transposed convolutional layers. The final output is an image that is conditioned on the preferences of Moira and Ferdinando.
3. **Batch Normalization**: The model employs batch normalization after specific layers to stabilize and accelerate the training process.
4. **Debugging**: The model includes a debug option that, when activated, prints the shape of the tensors at various stages of the forward pass. This is useful for understanding the flow of data within the model and ensuring that tensor dimensions align correctly.

In summary, `CVAE_ResNet18_DualEmbedding` is a sophisticated model that combines the power of CVAEs and ResNet18 to generate images tailored to specific conditions, in this case, the preferences of two respondents.

2.4 Benefits

The architecture amalgamates the advantages of VAEs and ResNet. VAEs excel in generating diverse and high-quality samples, making them apt for our image generation task. The conditional aspect, steered by reviewer scores, facilitates controlled generation. Lastly, the ResNet18 backbone, renowned

Overall, the transition from basic convolutional layers to ResNet18 architecture has significantly contributed to the effectiveness and versatility of Conditioned Variation Autoencoders, making them a powerful tool for tasks requiring customized and contextually guided image reconstruction.

We preferred CVAE as it has an explicit latent space which means they provide a structured and interpretable representation of data, CVAEs can inherently support both encoding and decoding, making them suitable for tasks where you need to map data from the data space to the latent space and vice versa and it tends to be more stable during training compared to GANs, which can sometimes suffer from mode collapse or training instabilities. GANs are known for their sensitivity to data and can be challenging to train, especially when dealing with high-dimensional data or data with complex distributions.

The code initializes a ResNet18 model with pretrained weights. The optimizer is initialized with a learning rate of 0.001. we have used the beta term for the KL-divergence loss. we also tried experimenting with using the MSE loss and MS-SSIM loss as the reconstruction loss.

Our chosen architecture for the task of conditional image generation is a Conditional Variational Autoencoder (CVAE) enhanced with the power of ResNet18. Here's an in-depth exploration:

2.4.1 ResNet18 Feature Extraction

The ResNet18 serves as our primary feature extractor for the images. Known for its deep structure and skip connections, ResNet18 efficiently learns intricate patterns and hierarchies in the image data. The features extracted from this stage have a dimensionality of 512, which corresponds to a flattened output from the ResNet18's last convolutional block.

2.4.2 Encoder

- **Input Concatenation**: Features extracted from ResNet18 are concatenated with the condition vectors, which are the reviewer scores. This ensures our VAE remains conditional, allowing for tailored image generation based on reviewer preferences.
- **Latent Space Representation**: The concatenated vector passes through two separate dense layers, outputting the mean (μ) and log variance ($\log \text{var}$) of the latent variables. These define the distribution from which our latent space sample will be drawn.

2.4.3 Reparameterization

This step enables gradient-based optimization through the stochastic sampling process. A random epsilon value is sampled, and using the computed mean and log variance, a sample from the latent distribution, z, is generated. This sample acts as our input for the decoding phase.

2.5 Embedding Generation and Score Handling

2.5.1 Purpose

The primary aim of the `generate_and_save_images_and_embeddings` function is to generate image embeddings using a trained model and save both the original images and the embeddings to specified file paths.

2.5.2 Procedure

1. **Image and Score Retrieval**:
    - A batch of images along with their associated scores for Moira and Ferdinando are fetched from the dataloader.

- The scores associated with each image are stacked along the second dimension to create a combined condition tensor, denoted as c.
- This condition tensor c is transferred to the designated device.

2. **Embedding Generation**:
   - The images, along with their condition vectors c, are passed through the model.
   - The model's `encode` function returns the embeddings for the images.

2.5.3 Notes on Scores

The scores for Moira and Ferdinando serve as condition vectors that potentially influence the embedding generation process. These scores are combined to form the condition tensor c, which is then used alongside the images during the encoding process. The condition tensor allows the model to be aware of the associated scores when generating embeddings, thereby making the embeddings potentially more informative or representative of the data.

The process ensures that the generated embeddings are influenced not just by the image content but also by the associated scores, allowing for a richer representation of the data. By saving these embeddings alongside the original images, it becomes easier to use them in downstream tasks, such as visualization.

2.6 Loss Functions

For our training process, we primarily utilized the Combined Loss which amalgamates the strengths of MSE, Perceptual, and Histogram-based loss. This choice was influenced by preliminary experiments which suggested that the combined loss yielded better reconstructions while maintaining training stability.

1. **MSE Loss**:
   - Measures the mean squared error (MSE) between the predicted and ground truth images.
   - **Pros**:
     - Easy to compute and differentiate.
     - Stable training due to its simplicity.
   - **Cons**:
     - May not capture perceptual differences effectively.
   - **Formula**:
     $$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(recon\_x[i] - x[i])^2$$

2. **MS-SSIM Loss**:
   - Measures the structural similarity between two images at multiple scales.
   - **Pros**:
     - Better captures perceptual and structural differences.
     - Often provides better visual quality in reconstructed images.
   - **Cons**:
     - Computationally more expensive than MSE.
     $$\text{MS-SSIM\_Loss} = 1 - \text{MS-SSIM\_Val}$$

3. **Perceptual Loss (VGG16)**:
   - Utilizes a pre-trained VGG16 network to compute the MSE between feature maps from an intermediate layer.
   - **Pros**:

  - Captures high-level semantic differences between images.
  - Often results in visually pleasing reconstructions.
- **Cons**:
  - Requires a pre-trained model, thus additional computational cost.
  - May not generalize well outside the VGG16 training data domain.

4. **Histogram Loss**:
   - Matches the histograms of the predicted and ground truth images.
   - **Pros**:
     - Ensures similar color and intensity distributions between predicted and target images.
   - **Cons**:
     - Only ensures similar distributions, not spatial structure.
5. **Combined Loss**:
   - A combination of MSE, Perceptual, and Histogram-based loss.
   - Allows leveraging the advantages of individual losses.
   - Weights (α, θ, λ) control the contribution of each term.
6. **KL Divergence**:
   - Measures the difference between the learned distribution in the latent space and a standard normal distribution.
   - Ensures continuity in the latent space, allowing for smoother interpolations.

$$\text{KLD} = -\frac{1}{2}\sum_{i=1}^{N}(1 + \log(\sigma^2) - \mu^2 - \sigma^2)$$

2.7 Training

1. **Optimizer**: The Adam optimizer is used with a learning rate of 1×10^{-3}.
2. **Loss Selection**: Depending on the chosen `loss_type` (e.g., 'mse', 'ms-ssim', 'combined'), the appropriate loss function is applied.
3. **Delayed Introduction of Losses**: To stabilize training, additional loss terms (like KL-divergence, perceptual, and histogram losses) are introduced after a certain number of epochs.
4. **Early Stopping**: If there's no improvement in the validation loss for a set number of epochs, training is halted.
5. **Latent Space Analysis**: Every 10 epochs, the correlation matrix of the latent vectors is computed and analyzed.
6. **Validation Loss Plateau**: If the validation loss doesn't decrease for a certain number of epochs, additional loss terms are gradually introduced.

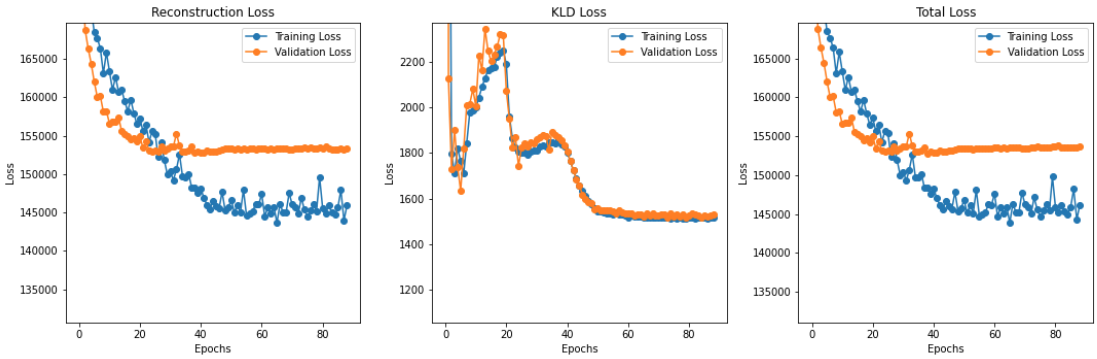
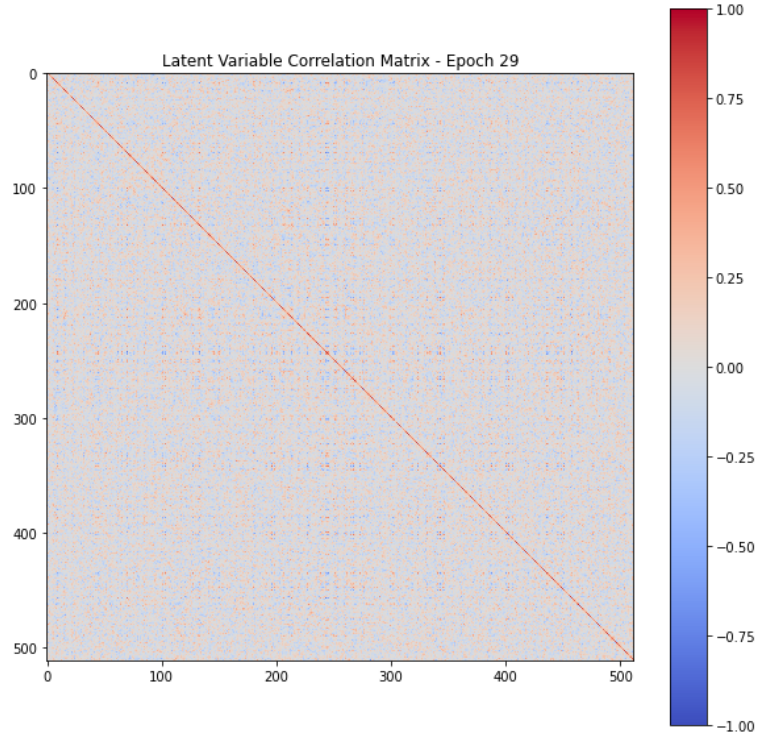


Figure 1: Training and validation losses using MSE and KLD metrics




```
Number of highly correlated pairs of latent dimensions: 38
Number of latent dimensions with low variance: 0

The latent dimensions seem to be reasonably independent, suggesting efficient utilization of the latent space.
Most latent dimensions have reasonable variance, suggesting they might be carrying significant information.
```

Figure 2: Latent space - monitoring of low variance and highly correlated pairs of latent dimensions

2.8 Comparison of CVAE, VAE-GAN, and VQ-VAE

2.8.1 CVAE: Conditional Variational Autoencoder

Advantages:

- *Controlled Generation*: By conditioning on certain variables, CVAEs allow for more controlled data generation.
- *Flexibility*: They can be used for various tasks like semi-supervised learning, multi-modal learning, etc.

- *Stable Training*: CVAEs generally inherit the stable training dynamics of VAEs.

Why CVAE for our use-case? Given that we're leveraging scores (from Moira and Ferdinando) to influence the embeddings and image generation process, CVAE is a natural choice as it allows the model to be conditioned on these scores.

2.8.2 VAE-GAN: Variational Autoencoder with Generative Adversarial Network

Description: VAE-GAN combines the strengths of both VAEs and GANs. The VAE component ensures a structured latent space, while the GAN component improves the quality of generated samples.

Advantages:

- *High-Quality Samples*: GANs are known to produce high-resolution, sharp images. Combining with VAE can yield samples of better quality than VAE alone.
- *Structured Latent Space*: The VAE ensures that the latent space has good properties, such as continuity and smoothness.

Drawbacks:

- *Training Stability*: GANs are notoriously difficult to train due to issues like mode collapse. Combining VAEs and GANs can sometimes exacerbate these training difficulties.

2.8.3 VQ-VAE: Vector Quantized Variational Autoencoder

Description: VQ-VAE uses vector quantization in the bottleneck layer to overcome the blurriness often associated with VAEs. This approach maps continuous encodings to a finite set of values, making it more discrete.

Advantages:

- *Less Blurry Reconstructions*: VQ-VAE can produce sharper images compared to traditional VAEs.
- *Hierarchical Structures*: VQ-VAE can be extended to hierarchical versions, allowing for multi-scale generation.

Drawbacks:

- *Complexity*: Implementing VQ-VAE can be more complex due to the need to manage codebooks for quantization.
- *Training Nuances*: Care must be taken during training to manage the balance between reconstruction and quantization loss.

Justification for Choosing CVAE: Given our use-case where we aim to condition the embeddings and image generation on specific scores, the CVAE offers the most straightforward and effective approach. It naturally allows for conditioning, which is central to our task. While VAE-GAN could potentially offer better image quality, the added complexity and potential instability during training might not justify its use. VQ-VAE, on the other hand, focuses more on addressing the blurriness in VAE reconstructions, which might not be as relevant for our embedding generation task.

3 Results

In this section, we present the results of generating car images based on Moira's and Ferdinando's scores using a Conditioned Variation Autoencoder (CVAE) with a ResNet18 architecture. The goal of this experiment is to leverage the CVAE model to produce customized car images that correspond to those scores.

3.1 Direct Model-based Generation

Generating car images appealing to Moira and Ferdinando using the trained model directly for each score. See Figure 6.

Pros:

- (List benefits of this method here.)

Cons:

- (List drawbacks or challenges faced with this method here.)

3.2 Ensemble Strategy

Generating car images appealing to Moira and Ferdinando using the trained model with an ensemble strategy.

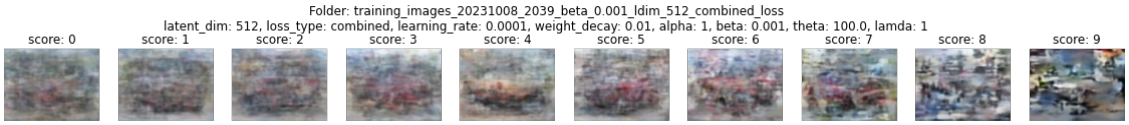


Figure 3: Car images appealing to Moira and Ferdinando using the ensemble strategy.

Pros:

- Easy to implement.
- No retraining required.

Cons:

- May produce blurry images due to averaging.

3.3 Probability Strategy

Generating car images appealing to Moira and Ferdinando using the trained model with a probability strategy.

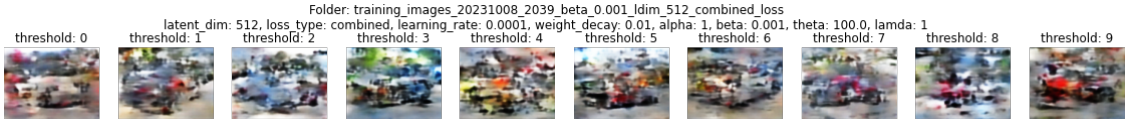


Figure 4: Car images appealing to Moira and Ferdinando using the probability strategy.

Pros:

- More flexible representation.

Cons:

- Might produce images that don't strongly correlate to any specific score.

3.4 Iterative Refinement

Generating car images appealing to Moira and Ferdinando using the trained model with iterative refinement.

Figure 5: Generate car images appealing to Moira and Ferdinando using Iterative Refinement

Pros:

- Sequential enhancement of images.

Cons:

- Requires multiple forward passes.
- Works a bit slower.

4 Discussion

Our endeavor to utilize scores from Moira and Ferdinando as conditioning factors in the CVAE model presented both challenges and insights.

4.1 Challenges and Insights

One significant challenge we faced was normalizing the scores. Properly conditioning a CVAE requires the additional data (in our case, the scores) to be effectively integrated into the model. There's a need to further explore this topic, either by modifying the model architecture or by reevaluating how we concatenate the scores along the dimensions.

4.2 Latent Space Insights

During our latent space analysis, where we monitored the correlation matrix of the latent vectors, we observed certain patterns and relationships between the variables. This analysis provided insights into how different image features and conditions might be interrelated in the latent space. However, a more in-depth study might be needed to derive substantial conclusions from this observation.

4.3 Hyperparameter Optimization

Our work also delved deep into hyperparameter optimization. Multiple hyperparameter search strategies were employed, including grid search, random search, and Bayesian optimization. However, our most optimal configuration was eventually discerned using the *Optuna* package, which helped fine-tune both hyperparameters and loss multipliers.

4.4 Future Improvements

There are several other techniques we could consider for integrating scores into the CVAE:

1. Dynamic Conditioning: Instead of a static concatenation, we can explore mechanisms where the model learns the best way to condition itself based on the scores.
2. Attention Mechanisms: By leveraging attention, the model could focus on specific parts of the score when generating or encoding images.
3. Adaptive Loss Function: The loss function could be adapted based on the scores, allowing the model to prioritize different aspects of the reconstruction based on the provided scores.

References

[1] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[2] Mathis Petrovich, Michael J Black, and Gül Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10985–10995, 2021.

[3] Adrian Alan Pol, Victor Berger, Cecile Germain, Gianluca Cerminara, and Maurizio Pierini. Anomaly detection with conditional variational autoencoders. In *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, pages 1651–1657. IEEE, 2019.

[4] Defu Qiu, Yuhu Cheng, and Xuesong Wang. Medical image super-resolution reconstruction algorithms based on deep learning: A survey. *Computer Methods and Programs in Biomedicine*, 238:107590, 2023.

Appendix

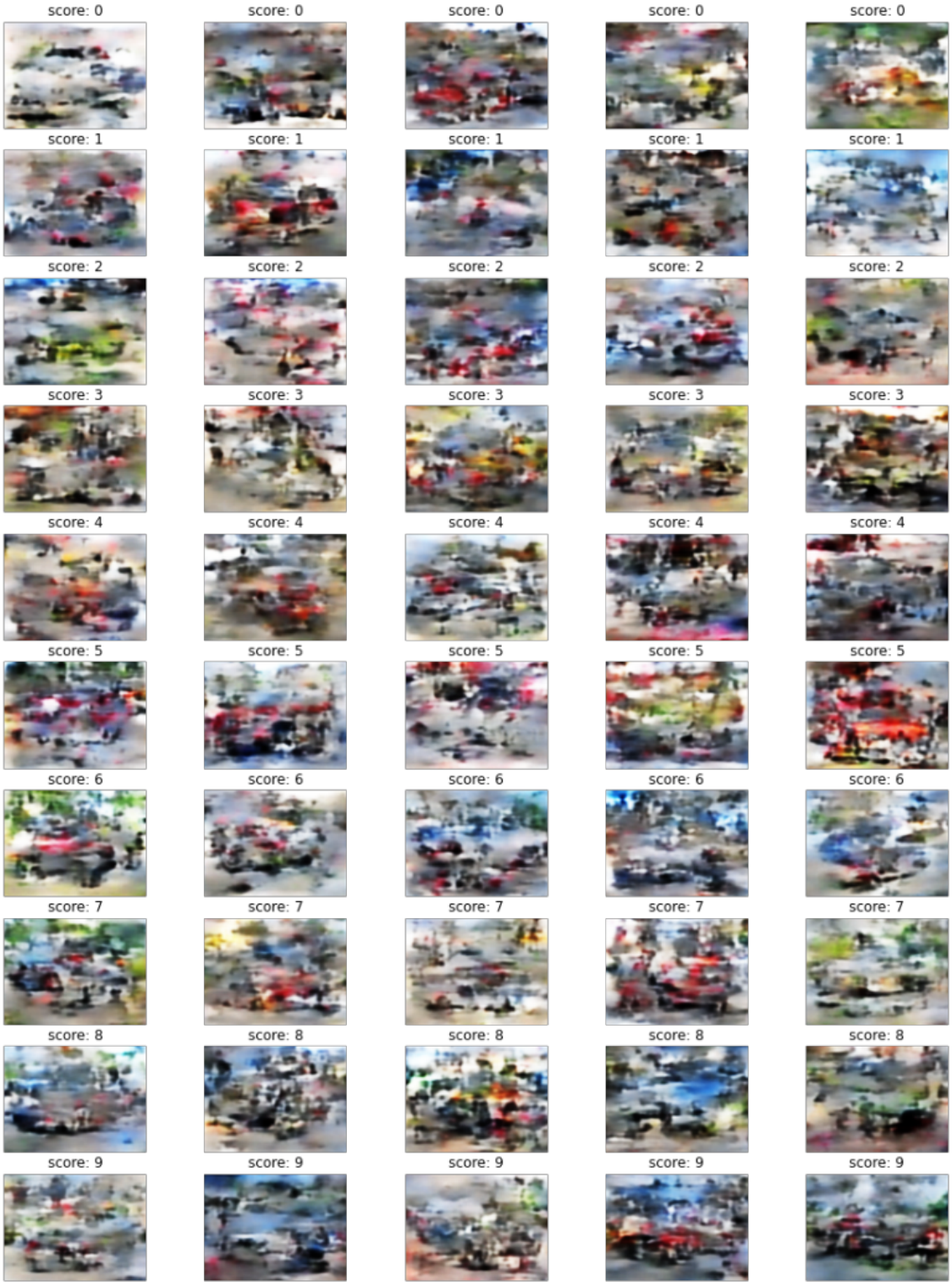


Figure 6: Car images appealing to Moira and Ferdinando using the model directly for each score.

Figure 7: Images generated with ResNet18 based CVAE model

Figure 8: Images generated with ResNet18 based CVAE model with batch normalization in a decoder