Synthesis of Mooney Images through a Generative Approach

Maria de la Paz Vives
Center for Data Science
New York University
mdv325@nyu.edu

Abstract

Mooney images are widely used in neuroscience because they offer a way to study visual perception. In addition to research, they constitute an important element in significant practical applications like face recognition and image based authentication. However, despite their relevancy, the number of actual Mooney images is very scarce and mainly limited to faces.
To address this limitation we propose a generative approach that allows for the automatic generation of high quality Mooney images, without the need of human intervention nor validation as existing approaches do. Moreover, we propose a mechanism that is not limited to faces and allows the generation of Mooney images of any type, from single objects to naturalistic settings, without the need to train any new models.

Keywords: *Perception, Visual Illusions, Mooney Generation, Autoencoders*

1 Introduction

A Mooney image is a degraded black and white image showing a single object that is very hard[1] or impossible to recognize without seeing the original grayscale image from which the Mooney was derived. However, after the subject is presented with the originating grayscale image (also called disambiguating image), an instantaneous recognition of the hidden object in the Mooney image occurs. Moreover, the observer won't be able to 'forget' it, i.e. he will not be able to see the Mooney image again without seeing the hidden object as it happens before the exposure to the original grayscale image.
To better understand this effect, take a few seconds looking at Figure 1 trying to identify the hidden object. Then, look at the corresponding originating grayscale image in Figure 11 and go back to Figure 1. You will now notice the hidden object on Figure 1 with no effort.

Figure 1: Example of Mooney Image.

Since their introduction (Mooney, 1957), images of this type have been actively used in different domains: from research to extremely practical applications like face recognition (Ke et al., 2017) and image based authentication (Castelluccia et al., 2017). However, there is only a limited number of hand-crafted images of this type and all the existing efforts, mainly based on image binarization, require manual evaluation because not every binarized image is a Mooney image.
Therefore, in our work we focus on the creation of Mooney images through a generative approach as an alternative method to binarization, with the end goal of building a mechanism to create high quality Mooney images without requiring human intervention, as the existing approaches do. In addition, we propose a framework where the definition of what a 'good Mooney' is can be defined and customized appropriately per use-case, without requiring the training of new models under different scenarios.

2 Related Work

In 1957, Craig Mooney published a set of 40 manually handcrafted human face stimuli to investigate perceptual closure (Mooney, 1957): the process of

[1] Recognition could take several seconds/minutes

visual completion based on prior perceptual knowledge. Even infants can recognize Mooney faces and how such a perceptual ability is achieved with sparse information may be key to understanding human face recognition in naturalistic settings, where noise, occlusion, and shadows are common (Ke et al., 2017).

Since their introduction, Mooney images have been widely used in the fields of computer vision and neuroscience, in particular for the study of human vision and perception. Imamoglu et al. (2012) and Arora (2023) utilizes these images for the study of conscious perception and object recognition. González-García et al. (2018) uses Mooneys to investigate the influence of priors in perceptual processing.

In addition, there has also been some research efforts purely focused on the generation of Mooney images given that the original dataset is very limited (only 40 human faces) and that it is often necessary to use a larger and more varied dataset for experiments and applications.

Ke et al. (2017) builds a Mooney face classifier as a way to evaluate black and white face images and pick up the most Mooney-like candidates. However, their approach is limited to faces and their definition of Mooney is subjective since they train the classifier on binarized images without human judgement (i.e. images that might not be Mooney). Similarly, Schwiedrzik et al. (2018) creates a dataset of 500 'Mooney' faces through a binarization approach where the threshold is subjectively set by the authors. Castelluccia et al. (2017) automatically generates two-tone images through a more sophisticated smoothing and thresholding process, but ultimately also requires human evaluation and filtering, an instrinsic limitation of the binarization approach.

Due to these limitations, we propose an alternative method to binarization for the creation of Mooney images. We suggest a generative approach and the use of a loss function which (1) objectively measures the quality of a good Mooney through its features (2) could potentially be customized per use case and (3) is independent from the training of the models.

3 Methods

In order to implement our proposal, we decompose it in the following two tasks:

Task 1 Train a model to reconstruct binary images in order to learn the latent space of these images. Note we intentionally select binary images for this task (vs. Mooney) because we want to learn the more general representation of blobs, including images in much more diverse and naturalistic environments than the existing Mooney images.

Task 2 Given any grayscale image generate the optimal[2] Mooney for it, using the latent representation space and decoder learned on the 1st task.

More formally, we are trying to generate an optimal Mooney given a grayscale image through the following optimization:

$$\hat{Mooney}|gray = G(\arg\min_{h} L(G(h), gray) \tag{1}$$

Where $gray$ is a grayscale image, h represents a sample of the binary images latent space, G is a generator function that given a representation from the latent space returns a Mooney image and L is a loss function that will measure the feature distance between the generated Mooney ($G(h)$) and the grayscale. Note that h and G are taken from Task 1. More precisely, h will be sampled from the distribution of the latent space learned during the training and G will be the decoder of the trained autoencoder.

It is very important to highlight that defining a good loss function L that is dependant only on the generated Mooney and the originating grayscale as we are proposing in Equation 1, is a complex and independent problem on its own (since there is no 'ground truth'). For this reason, for the scope of this work, we decide to use a simple function as a prototype (mean squared error) and leave the design of it for future work.

This decision is also aligned with our interest to keep the loss function as abstract and separate from the model training as possible in order to enable: (1) incremental design and experimentation of the loss function without the need of model re-training (2) customization of loss functions according to the scientists needs (e.g.: someone working on face recognition will have a set of specific patterns they want to focus on, while someone working on Mooney based authentication methods might

[2]Makes the most people experience the Mooney effect

favor much more generic images and patterns, as long as they are 'impossible' to recognize.) In short, the 'optimal Mooney' could potentially be different depending on the use case and we expect scientists to be able to use this framework for different purposes.
Lastly, as a consequence and drawback of this design decision, generating a Mooney vs. a black and white image will be fully dependant on the loss definition and therefore if the function is not appropriate this mechanism will not ensure Mooneys.

3.1 Dataset

As previously mentioned, differently to other Mooney studies that focus solely on faces, we are interested on the generation and study of the Mooney effect with naturalistic images. Therefore, we chose the Pascal VOC 2012 Detection Dataset (Everingham et al., 2010) for this work, where each image contains single or multiple objects in a diverse set of environments, as encountered in real life.
The dataset has 5717 images in the training split and 5823 in the validation split. The images are RGB and have different sizes, the majority of them are in the range of 400-500 pixels but there are other small images that are around 200x200 pixels. The splits are fairly balanced among classes (except for the person class) and there is a slight percentage of the images (around 15%) that contain multiple objects.

3.1.1 Transformations

Given that the original images are of different sizes and have three channels (RGB), key parts of our data pre-processing pipeline are the conversion to binary image (black and white) and the resizing to a common figure size, in addition to other convenient augmentations (like Gaussian Blur) that help make the model robust.
We can define our end to end data processing pipeline with three high level steps: (1) Resizing and conversion to Grayscale, (2) Augmentations and (3) Binarization, described in more detail in the following subsections. Examples of the resulting images (pair of black and white images) can be seen in Appendix A.2.

3.2 Resizing and Grayscale

We resized all images to a common figure size of 256x256. The resizing was done using bilinear interpolation and the new dimensions were chosen based on the size and range of the smallest images. In addition, we convert the RGB images to single-channel to obtain grayscale images with values in the [0-1] range. The output of this step is our dataset of interest: naturalistic, equally sized, grayscale pictures from which Mooney images are going to be generated.

3.3 Augmentations

We do smoothing to remove noise from the images through the application of Gaussian Blurring with a kernel size of (5,9) and sigma of (.1,.5), which define the amount and radius of the blurring correspondingly. It is in our plans to incorporate additional augmentations (RandomCrop, RandomFlip, etc.) as well as experiment with other common types of smoothing (e.g. average, adaptive) that have proven to be effective in practice.

3.3.1 Binarization

We initially approached the binarization by thresholding the grayscale image with a single value: all pixels above a particular luminance value were set to white and those below them were set to black. However, after noticing that many of the images were losing key shape information with this approach, we switched to a two-threshold binarization, where values above the upper threshold and below the lower threshold are set to white and the rest to black. Our resulting transformation has an upper and lower threshold of .6 and .2 respectively.

It is important to note that after this binarization process, we have some black and white images that are actual Mooney images (i.e. generate the Mooney effect), such is the case of 1 but many others that are not for the reasons explained in Figure 2. This helps illustrate why binarization as a method, without human evaluation, is not enough to create good Mooney images and a different approach, like the one we are proposing might be needed.

3.4 Modelling

As explained at the beginning this section, we decompose our problem in two different tasks. In Task 1, we train an autoencoder model to learn the latent representation of our binary images and in Task 2 we frame an optimization problem to find the hidden representation that generates the best 'Mooney' for a given grayscale.

Figure 2: Not every binarized image is a Mooney image: top binary image is recognizable without looking at the respective grayscale and bottom one is not recognizable even after looking at the grayscale since it barely resembles it.

3.4.1 Autoencoder Model

To learn the latent space of the black and white images, we train an autoencoder model to reconstruct these binary images. As shown in shown in Figure 3, the autoencoder is designed with 4 convolution layers for the downsampling (encoder) and 4 transposed convolution layers for the upsampling (decoder). The encoder compresses the input to a latent space of size 8x32x32 which is 12% of the original image size (1x256x256). Although not shown in Figure 3, each of the convolutional layers includes a Relu activation function followed a dropout layer with probability p.

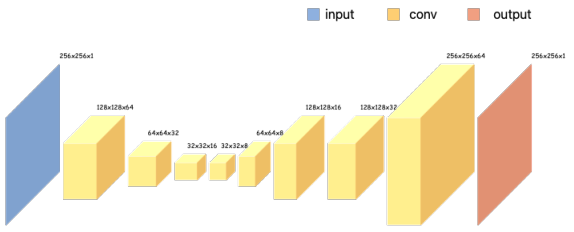

Figure 3: Autoencoder architecture for the reconstruction of binary images. Each convolutional operation is followed by a Relu activation function and a Dropout that are excluded from the diagram.

3.4.2 Autoencoder Training

We use the Adam optimizer with a batch size of 64 for all of our training experiments, setting a fixed seed of 0 for random generators for reproducibility purposes. We train our model with different values of learning rates and dropout probabilities.
In addition, we experiment with two different pixel level loss functions:

1. **MSE**: the model learns to predict pixel intensities and we calculate the loss by averaging the squares of the differences between the predicted and actual pixel instensity values.

2. **BCE**: the model learns to predict the probability of each pixel belonging to the class white (1) or black (0) and calculate the loss by measuring the distance between the probability distribution of the predicted vs. the actual value. Note that for practical reasons (numerical stability), the model outputs logits instead of actual probabilities and we calculate BCE with logits.

Regardless of the loss used during model training, model results are evaluated using accuracy, calculated as the proportion of pixels correctly classified with the true class. Given that the model will learn pixel intensities or logits depending on the choice of training loss, the outputs need to be converted to classes for the calculation of accuracy using the corresponding thresholds. In our case, this is .5 for an MSE trained model and .0 for a BCE trained model.

3.4.3 Generation of Optimal Mooney

To find the optimal Mooney image for a given a grayscale, we perform an optimization over the binary images latent space, i.e. look for the representation in the latent space that generates the best Mooney. More precisely, we:

1. *Sample from a normal distribution* with mean and std deviation as defined by the learned latent space (i.e. calculated using the encodings of all the binary images used during the training of the autoencoder).

2. *Define a loss function* that given a grayscale image and a sample from the latent space, calculates the 'feature distance' between the grayscale and the Mooney generated through the decoding of the sample.

3. *Find the hidden representation that minimizes the loss* as defined in the previous step.

As loss function for the optimization, we use mean squared error (MSE) between the grayscale image and the output of the model (logits and/or

pixel intensities depending on the model) and we evaluate our results on accuracy of the prediction (binary image) against a manually handcrafted Mooney that we take as 'ground truth'.

Observation: as explained in Section 3.4.3, to properly evaluate the generation of the Mooney images we would need an evaluation function based on features of the generated Mooney image and the originating grayscale. However, given that that effort is outside of our scope, we use a dataset of 20 pairs of grayscale and Mooney images designed for a human experiment, where the Mooneys have been human evaluated and therefore we believe that taking them as 'ground truth' to evaluate our generated images is an acceptable temporary approach for evaluation.

4 Results

4.1 Autoencoder

We train the autoencoder model for five epochs over all the combinations mentioned in Section 3.4.2 and we summarize the results in Table 1. We take as our baseline a random classifier that predicts 0 or 1 at a pixel level and achieves a **.5104** accuracy in our validation dataset.

Loss Fn	Dropout	LR	Accuracy
BCE	.3	.01	.8993
BCE	.3	.001	**.9059**
BCE	.5	.01	.8905
BCE	.5	.001	.8801
MSE	.3	.01	.8956
MSE	.3	.001	**.8972**
MSE	.5	.01	.8942
MSE	.5	.001	.8428

Table 1: Autoencoder Training Results. Best BCE and MSE trained models highlighted.

4.1.1 Training Observations

In Figure 4, we show the learning curves for all our experiments, with the charts on the top displaying the trainings done with binary cross entropy (BCE) as the loss function and the charts on the bottom showing those trained with mean squared error loss (MSE). As expected, as training progresses we can see the training loss decreasing and the accuracy increasing for all experiments, i.e. independently of loss function. At the same time, based on the accuracy plots we can note that the trainings with BCE, with a steady increase, looks more stable than the MSE ones with their peaks.

An additional observation when comparing BCE and MSE trainings is that, independently of their differences, the ranking of each training combination (Dropout+LR) is the same for each loss function. In other words, based on validation accuracy, the two different loss functions define the same performance order for the different combinations of training settings (dropout+LR). This fact, together with the ranking of combinations, can be easily visualized in Figure 5.

Last, doing a similar breakdown per dropout rate, shown in Figure 6, we can see that for any fixed combination of training loss and learning rate, dropout rate .3 outperforms the training with dropout .5, which suggests us that dropping lower in dropout might bring further improvements.

4.1.2 Reconstructions Observations

We present a sample of a reconstruction done by the best autoencoder model (BCE, LR=.001, Dropout=.3) in Figure 7, with the original grayscale and corresponding binary image (input to autoencoder) on the top and the reconstruction done by the autoencoder on the bottom.

First, we can observe the high quality of the reconstructed image (bottom right), with very slight differences from its input (top right).
Second, as it can be seen on the bottom left image, the grayscale reconstruction show grid-like artifacts that we observed in all the variations of the autoencoder. In order to improve this, we experimented doing an architectural change to the model shown in Figure 3: we added a convolutional layer after each deconvolution in the decoder. The addition of these layers usually help the decoder correct undesired artifacts, in this case the grid pattern.
Although we could note a difference in the resulting reconstruction, the model required much more training to converge due to the additional layers and we could not see a significant difference on the resulting binarized image (i.e. the validation accuracy was the same). For that reason, we did not extend the analysis on that model nor we are reporting results at this time. However, in future work, we want to revisit this direction as it might make a difference on the hidden representation and therefore, it could have an impact on our second task: generation of the optimal Mooney from the latent space.

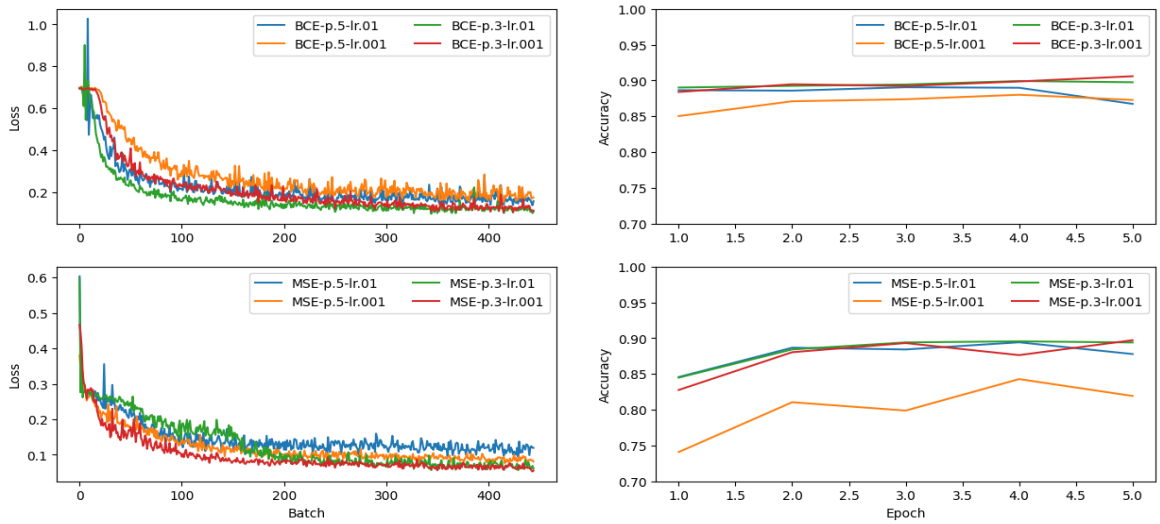

Figure 4: Training loss (left) and validation accuracy curves (right) across 5 epochs of model training. Figures on the top correspond to learning with BCE (Binary cross entropy) loss function and those on the right with MSE (mean squared error).

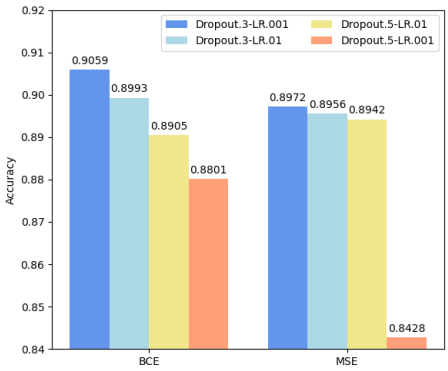

Figure 5: Accuracy of training settings broken down by loss function. The same performance ranking (from left to right) can be observed for both loss functions.

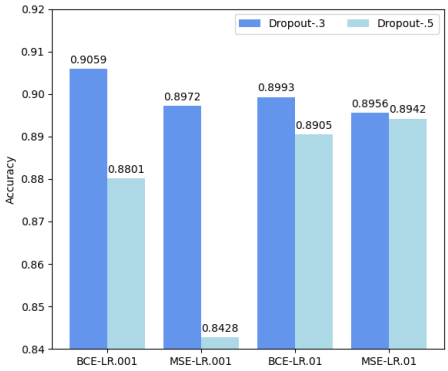

Figure 6: Training accuracy for each combination of loss and learning rate, broken down by dropout. Lower dropout outperforms higher rate given the same loss and learning rate settings.

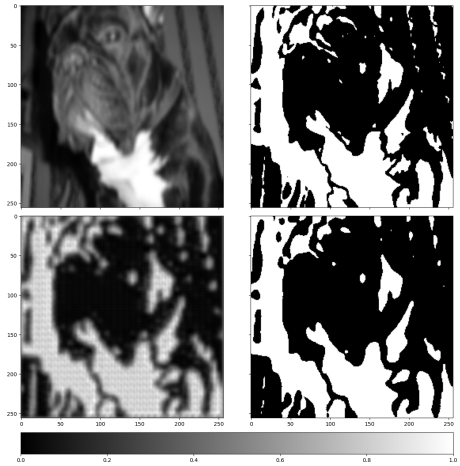

Figure 7: Reconstruction by best model. The originating grayscale is on the top left and its corresponding binary image to its right (autoencoder input). In the bottom, the model logits on the left and the binary predictions on the right.

4.2 Generation of Optimal Mooney

We generate the 'optimal' Mooneys through the procedure described in Section 3.4.3, for a set of 20 grayscale images for which we have the 'ground truth' Mooney (binary image).

We take the decoders from the best BCE and MSE models (highlighted in Table 1) to generate a Mooney from a sample of the corresponding latent space and run the optimization using stochastic gradient descent. We do 1000 iterations for the optimization process due to resource constraints, but longer optimization would be desired since we

can see that, for many images, the accuracy is still increasing after this number of iterations.

4.2.1 Aggregated Results

We display the average accuracy across the 20 images for each model in Table 2 and show some examples of generated binary images in Figure 8.

Model	Optimizer	Iterations	LR	Accuracy
BCE	Adam	1000	.1	.7427
MSE	Adam	1000	.1	**.7683**

Table 2: Average accuracy of generated Mooney across 20 images, calculated against ground true Mooneys (manually handcrafted).

Given that a random generation of a Mooney would be around .5 of accuracy, these results are very promising as our first approach, specially considering that for many of these images the optimization had not reached its minimum by the time it was stopped. Moreover, the images shown in Figure 8 reflect the potential of the approach generating the desired binary images.

Figure 8: Examples of generated Mooneys (second row) from grayscale images (first row) through the proposed optimization method using the BCE model.

4.2.2 Image Level Results

In Figure 9, we show the accuracy for each of the images we used for the aggregated evaluation.
We can see that taking accuracy as the evaluation metric, there is no absolute best model for every image, i.e. for some Mooney generations BCE outperforms MSE and viceversa. However, we see more cases where MSE is strongly more accurate than BCE, which makes us think that it might a better model for the task (also supported by the overall accuracy presented in the previous sectionn).
This is further reinforced by looking at some of the generated pictures, where the MSE generated image is much cleaner than the BCE one - even when the accuracy is the same (Figure 10).
These observation only highlight the complexity of the evaluation of these images and shows the first significant limitations of our work at this phase, where we decided to leave the definition of the loss and evaluation function outside of this work and used simplified functions (MSE and accuracy) as prototypes to setup the framework.
On the positive side, the generated images and overall results prove that the proposed generative mechanism is feasible and constitutes a valid approach to generate Mooney images as long as we are able to define a good loss function based on the Mooney expected features with respect to its originating grayscale.

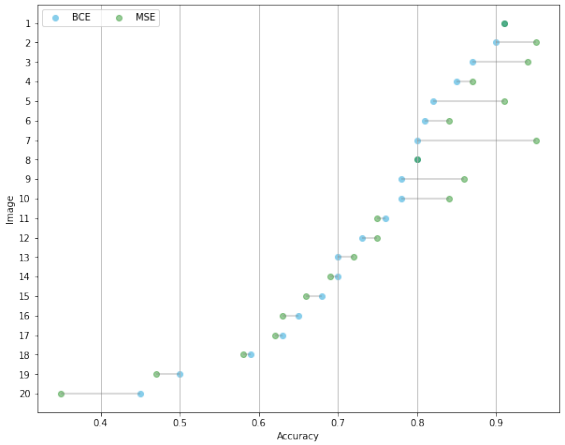

Figure 9: Accuracy comparison per grayscale image (20) on generating the optimal Mooney from the grayscale using the best BCE and best MSE models. Grayscale images come from a human experiment dataset and therefore accuracy is calculated against ground-truth Mooney images (i.e. human handcrafted).

5 Conclusion

We prepared a large dataset for the training of an autoencoder that reconstructs binary images to learn their latent space. We successfully trained the autoencoder with it and learned insights from the different training configurations (loss functions, learn-

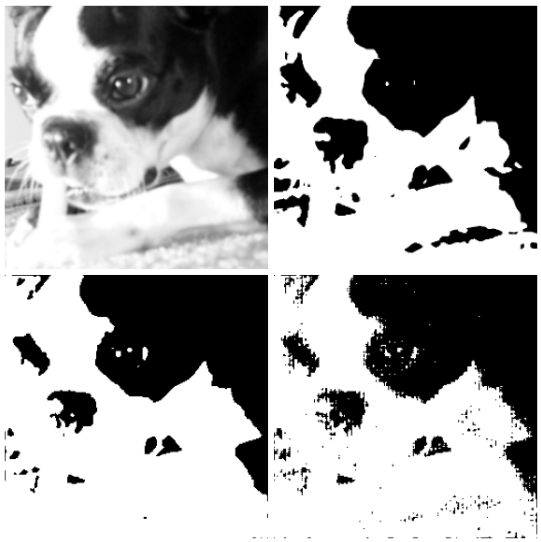

Figure 10: Mooneys generated through the optimization method. The images on the top are the originals from the human dataset (grayscale and mooney) and the two in the bottom are the ones generated by the MSE (left) and BCE (right) models given the original grayscale. This image corresponds to Image 1 in Figure 9, which has the same accuracy (.91) for both models.

Figure 11: Originating grayscale image for Mooney in Figure 1

ing rates and dropouts). With this initial approach we outperformed the baseline by a significant number (.9 vs .5 accuracy).

Next, we took a human evaluated dataset with real grayscale and Mooney images and we implemented the optimization framework for a generative approach to Mooney images. The results of our initial effort (.77 accuracy) together with the high quality individual images we are able to create proves that the method we propose to create Mooney images is a valid and promising way towards the generation of quality Mooney images in scale, without the need of human intervention and/or evaluation. Moreover, the approach allows for a customization of a loss function depending on the use case and does not require retraining of any models.

In the immediate future, we plan to work on a definition of a loss function that only depends on features of the original grayscale and the generated Mooney because we believe that is the missing component for a proper evaluation of the generated images and the generation at scale.

6 Contribution statement

This work has been fully developed by the single author as an attempt to contribute to the study of perceptual processing at Eric Oermann's Lab (OLAB) at NYU Langone under the supervision of Phd student Chris Xujin Liu.

References

Riya Arora. 2023. Understanding human perception through mooney faces.

Claude Castelluccia, Markus Duermuth, Maximilian Golla, and Fatma Deniz. 2017. Towards Implicit Visual Memory-Based Authentication. In *Network and Distributed System Security Symposium (NDSS)*, San Diego, United States. ISOC.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2010. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338.

Carlos González-García, Matthew W Flounders, Raymond Chang, Alexis T Baria, and Biyu J He. 2018. Content-specific activity in frontoparietal and default-mode networks during prior-guided visual perception. *eLife*, 7:e36068.

Fatma Imamoglu, Thorsten Kahnt, Christof Koch, and John-Dylan Haynes. 2012. Changes in functional connectivity support conscious object recognition. *NeuroImage*, 63(4):1909–1917.

Tsung-Wei Ke, Stella X. Yu, and David Whitney. 2017. Mooney face classification and prediction by learning across tone. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2025–2029.

C. M. Mooney. 1957. Age in the development of closure ability in children. volume 11(4), page 219–226.

Caspar Schwiedrzik, Lucia Melloni, and Aaron Schurger. 2018. Mooney face stimuli for visual perception research.

A Appendix

A.1 Code repository

All the code used for this paper is publicly available at `https://github.com/pazvives/mooney`.

A.2 Binary VOC Dataset

Figure 12: Examples of the Binary VOC Dataset that is generated from applying the data transformations explained in Section 3.1. On the left we have the resulting grayscale images and on the right, we have the binary images used to train the autoencoder. Note that the later ones are black and white images but not necessarily Mooney.