Enhancing Neural Network Explainability with Variational Autoencoders

Loc Tran and Chester Dolph
NASA Langley Research Center, Hampton, VA, 23666

Derek Zhao
Columbia University, New York, NY 10027

Machine intelligence has been used to tackle increasingly complex problems and deep learning solutions are at the forefront of tackling these problems. In general, these architectures have a great number of parameters that are methodically updated in training. The vast number and complexity of deep neural networks makes it very difficult to decipher the inner workings of the neurons and layers that make up the network. This paper posits that trustworthiness and trust in autonomous systems are increased through eXplainable Artificial Intelligence (XAI) and presents a method that enhances the explainability and understanding of a neural network decision. We leverage variational autoencoders to produce human interpretable features from complex data sets. We show that the explainable features can then be used for machine learning applications. Explainability inspires trust in autonomous systems that use deep learning, which is necessary for safety critical systems.

I. Introduction

Image-based deep neural networks (DNNs) are often used as a black box where users train the network using large image data sets, supply an input image, and during inference time, a decision is made with little supporting evidence to why the decision is made. A DNN is composed of multiple layers between the input and output. Researchers have been tuning DNNs layers and architectures to get improved classification accuracy and speed for their applications while forgoing explainability of the system. Despite the success of DNN and their improved architectures, the internal mechanisms of what causes the high classification accuracy remain unclear. Much of DNN research has been concentrated on improving classification accuracy, training time, and inference speed of neural networks without focusing on the explainability of the inner layers. But, neural network interpretability is a burgeoning area of interest [1–5]. A better understanding of the internal networks may lead to improved public trust of DNNs to perform safety critical tasks and improved robustness to adversarial attacks [6]. Deep network interpretability is a challenging problem because the internal components are non-linear representations of 2D images at varying levels of feature extraction with complex patterns that are visually not intuitive [7]. The goal of this work is to achieve a greater understanding and explainability of DNN to enable effective teaming of humans and machines where establishing trust between agents executing a shared mission. Our contributions include showing that it is feasible to generate human explainable features from a neural network using variational autoencoders and that these features provide some justification of machine learning decisions, but it is currently not feasible to rely on these explainable features in critical systems.

II. Background

Previously, we explored if there was selectivity of the neurons in a pre-trained neural network [7]. We focused on GoogLeNet [8], a popular image classification neural network which had state of the art classification performance on the ImageNet data set [9] when it was released. This particular network has approximately 6.8 million parameters. With the vast number of parameters, a few questions can be posed. Is there a neuron that is selective to a particular feature of an object? For example, is there a neuron that is selective to cat's ears that is used when detecting and identifying cats?

In this paper, we explored ways to induce selectivity by training a variational autoencoder (VAE) with constraints. It has been shown in the literature that disentanglement of human describable features could be achieved with a simple modification of a VAE [10]. The approach was then to replicate these results and analyze the use of these features. We were able to extract explainable features from a synthetic data set and also a data set of human faces. Later, the disentangled features are used in machine learning tasks.

The overall goals of this effort is to be able to produce explainable features from an AI system to justify decisions and show that explainable features could still be used in machine learning applications rather than unconstrained features that are trained by conventional neural networks with the focus of potentially leveraging these explainable features to certifiable algorithms for small unmanned aerial systems.

III. Algorithm

A. Variational Autoencoders

An image autoencoder [11] can be seen as a neural network that projects the input data into a hidden latent space and then projects the latent space back into the original image space. The training error function is a similarity function such as the squared sum of errors (SSE)

$$SSE = \sum_{i}^{n} (x_i - y_i)^2$$

The first half of the autoencoder is referred to as the decoder which translates an image to the latent space. Similarly, the second half of the autoencoder converts the latent space's code back into image space. The latent space of could be seen as a dimensionality reduction of the original data set. We previously had found that reprojecting neural network layer and weight activations into image space helped in interpretation of the layer[7]. Similar to conventional neural networks, slight modification of values in an autoencoder's latent space could correspond to large deviations in the generated image and are difficult to analyze.

The variational autoencoder (VAE) was introduced by Kingma et al. [12]. The scalar latent features are now viewed as probability distributions. Most instances of variational autoencoders model the latent distribution as a Gaussian $\mathcal{N}(\mu, \sigma)$. Each latent feature in the latent space is represented by one neuron for the mean, μ_j, and one neuron for standard deviation σ_j.

Probability distributions are not used as nodes in a neural network because there was no way to backpropagate derivatives of a probability distribution. VAE's get around this using the so called reparameterization trick which expresses a gradient of an expectation into an expectation of a gradient. During the training phase of the VAE, the distribution is sampled from the probability distribution.

$$\mathcal{L}(x, y) = SSE + \mathrm{KL}(q_{\mu,\sigma^2}(z|x)||\mathcal{N}(0, 1))$$

where KL is the Kullback-Leibler divergence.

The latent parameters of the VAEs have interesting properties. The latent parameters model a continuous probability distribution. Small perturbations of a latent parameter makes a small change in image space. Latent traversals could now be applied where all nodes in the latent space are kept constant except for one. The effect of that particular latent feature could be isolated and viewed in image space by generating images by feeding the latent features through the decoder network.

Higgens et al. introduce the β-VAE with a simple modification to the cost function [10]. A single weighting parameter,

β is applied to the KL divergence term.

$$\mathcal{L}(x, y) = SSE(x, e) + \left[\beta[\mathrm{KL}(q_{\mu,\sigma^2}(z|x)||\mathcal{N}(0, 1))\right]$$

where $\beta >= 1$. It can be seen that when $\beta = 1$, the network is a standard variational autoencoder. And when $\beta > 1$, it is apparent that more weight is applied to the second term of the cost function.

The Kullback-Leibler (KL) divergence term measures how one probability distribution is difference from another probability distribution. The prior used for the latent neurons is typically the standard Gaussian, $\mathcal{N}(0, 1)$. So the higher weight on the KL divergence could mean that Gaussian distributions that the loss function is sacrificing reconstruction error for distributions that more closely match a Gaussian distribution. This causes a more disentangled latent space which also corresponds to the neurons having more selectivity. Other researchers have attempted to extend this work to understand the disentanglement [13], improve the reconstruction accuracy [14], and applying the network to semi supervised learning [15].

The introduction of the β hyperpameter as a scaling factor of the KL divergence presents a balancing act. There are two goals of the loss function, the first is reducing the reconstruction loss and the other is related to the adherence of the latent parameters to a Gaussian distribution which is the KL-divergence term. A low β induces low disentanglement while having better reconstructions. A high β results in good disentanglement but the reconstructions are generally blurrier. Therefore, the β-VAE tends to have worse reconstruction accuracy compared to standard VAEs and other image generation networks such as generative adversarial networks.

Chen et al. improves on the reconstruction accuracy with the β Total Correlation VAE (β-TCVAE) by breaking down the KL diverenge term to three terms, mutual information, total correlation, and dimension-wise KL [14]. The β weight is only applied to total correlation which induces disentanglement while the other terms are not weighted.

IV. Evaluation

Precision-recall was the primary evaluation method for the different reconstruction loss functions evaluated in this work. In this section, we describe experiments to evaluate the level of disentanglement for two image data sets. A visualization tool is also described that was used to aid in assessments of disentangled VAE models. An anomaly detection experiment was conducted to detect out-of-distribution images. Finally, we describe a toy experiment where VAEs could potentially be leveraged for person identification.

A. Disentanglement Experiments

Two data sets were in evaluation for this work: dSprites and CelebA. dSprites has been previously used by [10] for disentangling features in latent space. A sample of the images in dSprites is shown in Fig 1. The images are generated using five parameters for shape, scale, orientation, x position, and y position. The shape can be square, ellipse, or heart. The intrinsic dimensionality of this data set is known to be the five generating parameters so a perfectly disentangled representation would extract these parameters into its latent space.

A VAE and β-VAE were trained on the data set. Fig 2 shows the latent traversals for the VAE and Fig 3 shows the latent traversals for the β-VAE. For the latent traversal, all of the latent parameters are fixed except for one. In the Figures, each row corresponds to a different latent variable. The variable parameter is sampled from $[-3, 3]$ and the latent parameter set is passed through the generator network to produce an image. The progression along this traversal is shown by looking left to right on the columns of Figs 2 and 3. In Fig 2, the entire latent space of the VAE is entangled together. For example, the top row shows one latent parameter that is controlling scale, shape and position of the shape.

The latent traversals of the β-VAE are shown in Fig 3. The rotation, x-direction, and y-direction is successfully disentangled. The scale appears to be properly disentangled for most of the traversal but the shape transitions from an oval to a square at the end of the range. Similarly, a separate neuron is also controlling rotation and is entangled with a shape transition. The reason for the shapes being entangled may be because it is a discrete parameter while the

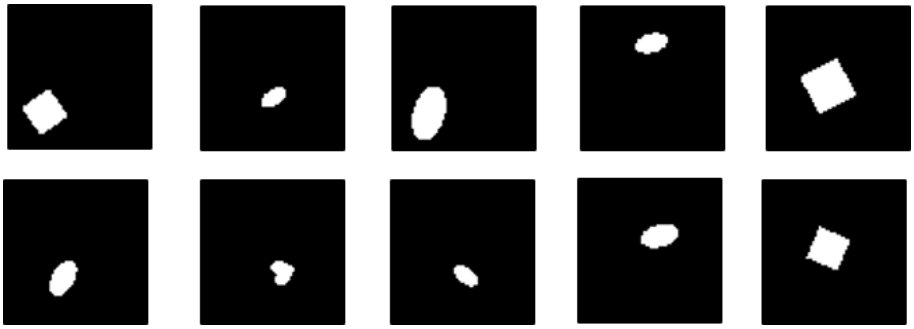

Fig. 1 DSPRITES example image

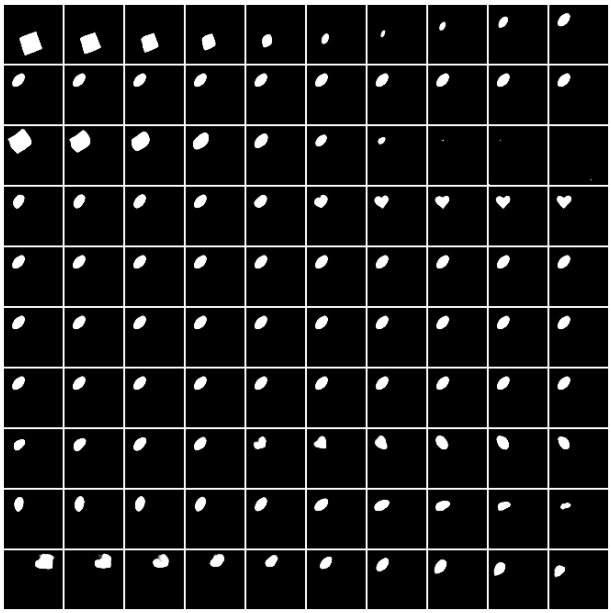

Fig. 2 dSprites VAE traversal.

parameters that are disentangled are continuous. Recall that the KL-divergence term is adding a constraint that the latent parameters model Gaussian distributions. Since the discrete shape classes are not modeled well as a Gaussian distribution, the VAE finds it difficult to represent it in latent space.

The second data set used to analyze the VAE was CelebA [16]. This consists of 200 thousand images of celebrity faces. The motivation of using this data set is that human faces contain a lot of variability. Furthermore, the expected latent space in the VAE is not known unlike the dSprites data set. When training the CelebA data set using 32 latent features, fourteen of the latent traversals produced significant changes of the facial image which are shown in Fig 16 - Fig 29. Each latent variable is shown in a different figure. The rows in these figures correspond to a different seed image and looking left to right of the columns are sampled images from the latent traversal.

Many of the variables in the β-VAE latent space produced disentangled features. For example, Fig 16 shows bottom to top rotation, Fig 18 shows left to right rotation. Several the disentangled features related to hair variations such as Fig 20 which is selective to the location of the hair part, Fig 21 is selective to the hair length, and Fig 22 which is selective to the hairline. These features were considered mostly disentangled because slight variations in the person's face are present in the latent traversal even though the latent parameter controlled a significant facial feature.

Two of the latent parameters were related to a person's smile but were also entangled with other facial features. Fig 28 shows a latent feature that controlled the level of smile but was entangled with skin tone and facial hear. Similarly Fig

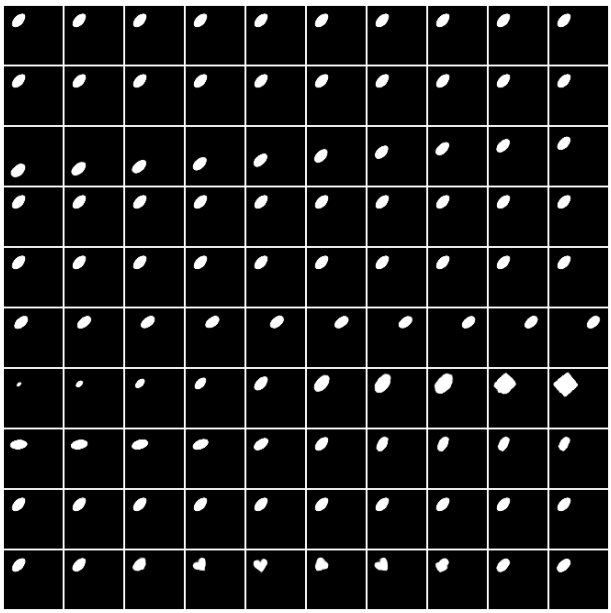

Fig. 3 dSprites β-VAE traversal.

29 shows the entanglement of smile, face width, and skin tone. Besides the neutral and smiling face, no other facial emotion was extracted.

The remaining 18 latent features showed little to no effects on the generated image.

B. Live Visualization Tool

A user interface was developed to visualize the latent features of the VAE in real time time. Figure 4 shows the visualization tool. On the left panel, an image of a person's face can be loaded from a file or streamed from a webcam. An overlay is provided so that the image can be aligned such that the eyes, nose, and mouth are in the same positions as the training data. There capture button freezes the live image from the web camera.

The center panel displays a list of latent variables. The description is labeled by a human description of the variable. The input image is passed through the encoder portion of the VAE to output the latent values.

The right panel shows the image produced from the latent parameters when passed through the decoder network. Reconstruction loss metrics are displayed below the reconstructed image.

C. Anomaly Detection Experiment

An experiment was conducted to see if the disentangled features from a VAE could be used in a simple machine learning problem. A classification problem was conducted to see if the VAE features could be used to determine if input images were faces or non-faces. Here, a β-TCVAE was trained on faces from the CelebA data set with 1000 faces left out of the training set for testing. The CelebA data set was augmented with 100 non-face images which were extracted from the COCO data set for anomaly detection.

Features that could be used in this classifier are not only the latent encoding but also the reconstruction error. It is expected that the VAE would likely not be able to reconstruct the image if it is not a face image and the reconstruction would have a high reconstruction error. The classifier is not limited to the particular reconstruction loss function that the neural network was trained on. Six reconstruction loss functions were used to evaluate the quality of the reconstructed

Fig. 4 Latent visualization tool.

image compared to the original input image:

- Mean Square Error (MSE) measures the mean sum of squared errors for the input and output.
- Mean Absolute Error (MAE) measures the mean sum of absolute values of the input and output.
- Cosine Distance is the dissimilarity of the inner product of the input and output image.
- Structural Similarity Metric (SSIM) is a perception-based metric initially used to measure image degradation [17].
- Cross entropy measures the sum of binary cross entropy between all combinations of channel pairs between the input and output.
- Color cross entropy measures the sum of binary cross entropy between red-red, green-green, and blue-blue channel pairs between the input and output.

For the initial experiment, logistic regression was used as the classification algorithm using only the six reconstruction losses. It was conducted 100 times using a random sampling where 70 percent of the data was used for training the logistic regression and 30 percent was used to test. The precision-recall curve for the classification problem is shown in Fig 5. Precision and recall is defined as below:

$$\text{precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$\text{recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

One way to think about the difference between precision and recall is that perfect precision means that there are no false positives. In this case, it would mean the algorithm would not characterize a non-face as a face. Similarly, perfect recall would mean there are no false negatives. In this case, a perfect recall classifier would not report a face as a non-face. A precision-recall curve is useful in analyzing two-class classification problems because it shows the relationship between false negatives and false positives. A perfect precision-recall curve would be a constant 1-value throughout the curve. Precision-recall curves are useful for evaluating the balance of true positive rate and false positive rate in machine learning models.

While the average precision of this classifier is fairly low, note that the number of non-face images was also very low

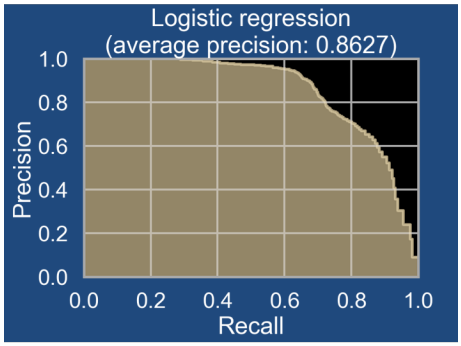

Fig. 5 Precision-Recall curve.

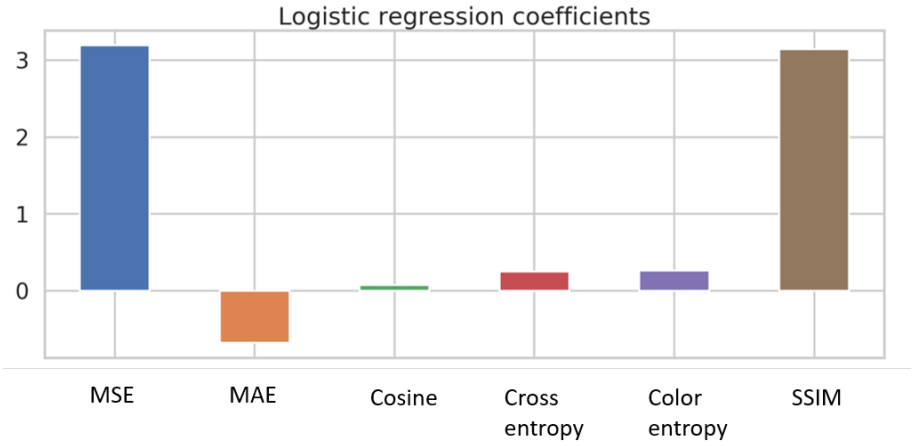

Fig. 6 Coefficients of logistic regression.

which means that this classifier is likely to be under-trained. Further analysis was done on the results of this classification problem. The coefficients of the regression function is shown in Fig 6. It can be seen that MSE and SSIM have the largest coefficients and are more important to the classification compared to the other reconstruction losses.

An interesting question that arises is, what images cause the classifier to fail and what faces are classified as non-faces? The images were sorted by order of mean square error reconstruction loss and the extremes of the classifier were analyzed. Figs 7, 8, 9, and 10 all show the original face and non-face image image on the left while their reconstruction is on the right. From the non-face input images, it can be seen that the image from the generator always has a face in it. A commonality of the non-face images in Fig 7 are that the images tend to lack a lot of detail. There is generally a large portion of the the input image that has low texture with one or two colors. The face images that are generated result in a low mean square error because they match the large areas of low texture. The low texture can be seen in the face images with the best reconstructions which are shown in 9. These input images have very little background texture which results in a low MSE. The non-face images with the highest mean square error are shown in Fig 8. The input images generally have a high degree of texture which the VAE cannot reproduce.

From this analysis, it was determined that the MSE reconstruction loss placed a great deal more weight on large swaths of pixels that are similar in color rather than small areas of high detail. For the face data set, the structural similarity metric was selected since it more closely aligns with how a person judges the similarity of two images. The β-TCVAE was retrained using SSIM during the training phase and the experiment was repeated. Fig 11 shows a side-by-side comparison of the precision recall curve between training using cosine distance and SSIM. The β values here are different because each loss function required a different β to produce disentanglement of the face data set. There is a significant improvement using SSIM as the loss function.

Fig. 7 Non-face images with the lowest mean square error.

Fig. 8 Non-face images with the highest mean square error.

Fig. 9 Face images with the lowest mean square error.

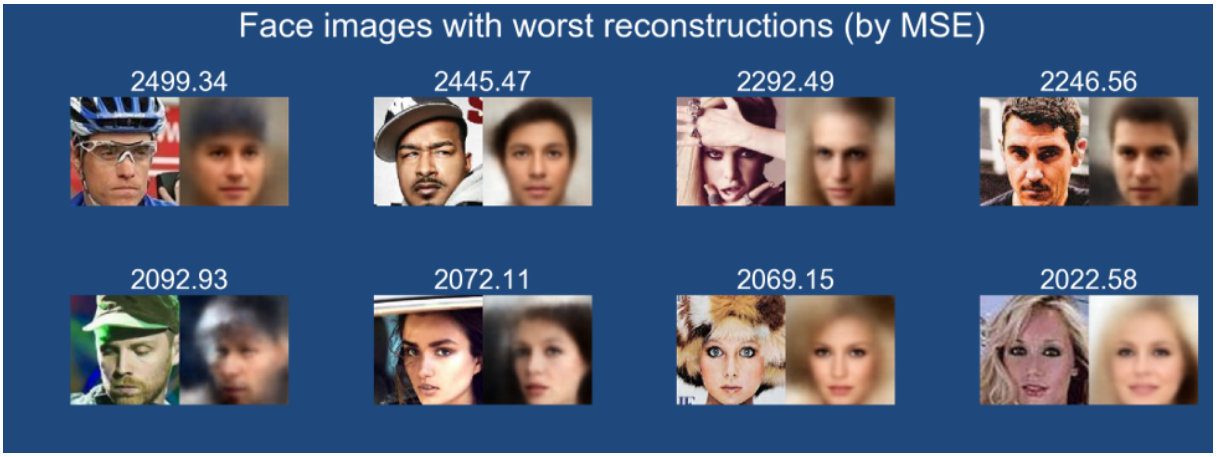

Fig. 10 Face images with the highest mean square error.

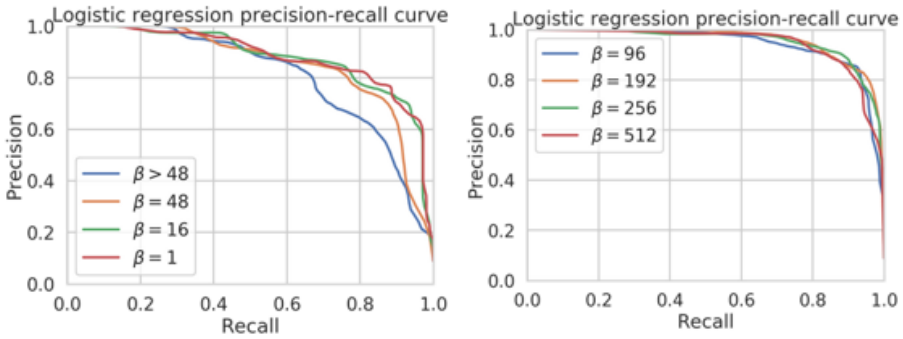

Fig. 11 Precision recall curve using varying levels of β (Left) Using cosine distance loss function. (Right) Using SSIM reconstruction loss function.

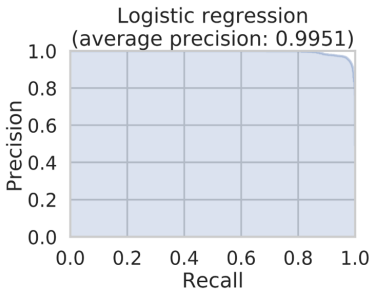

Fig. 12 Precision-Recall curve using 1000 non-faces for training

Fig. 13 Simulated person for search and rescue application.

The previous experiments were conducted using only 100 non-face images for training. Furthermore, only 70 images were used to train the classifier which proved to be too low to optimize the classification accuracy. This allowed us to see what images would cause the under-trained classifier to fail and a method to improve performance. The experiment was conducted using 1000 non-face images for training. Again, logistic regression was used as the classifier. This time, the latent parameters along with the reconstruction losses were used as features. The precision-recall curve is shown in Fig 12. The fully optimized logistic regression produces a highly accurate and precise classifier on this data set. This shows that the latent parameters and reconstruction losses of the disentangled data set are potentially useful in machine learning tasks.

D. Toy Experiment

One of the disentangled features that was extracted by the β-TCVAE was the presence of sunglasses which is shown in Fig 26. This experiment was to compare the selectivity of the neuron that measured the presence of sunglasses with a face that was not in the training data set to simulate a search and rescue scenario where an image of the missing person is available. A computer generated model was placed in a simulation environment to generate the sample images which are shown in Fig 13.

The β-TCVAE was applied to both the original image and the target image with the sunglasses. Fig 14 shows the latent encoding of the two images. The blue bars correspond to the original image while the red was from the target image.

To see the comparison better, Fig 15 shows the difference between the latent encodings. It is evident that difference of latent neuron z_{17} is disproportionately greater than all others. This was of course the neuron with selectivity over sunglasses.

This shows that latent space comparisons of disentangled neurons could potentially be used directly. The β-TCVAE could be able to detect that the target face and the original missing person's face were similar and that the difference corresponded to a non identifying feature.

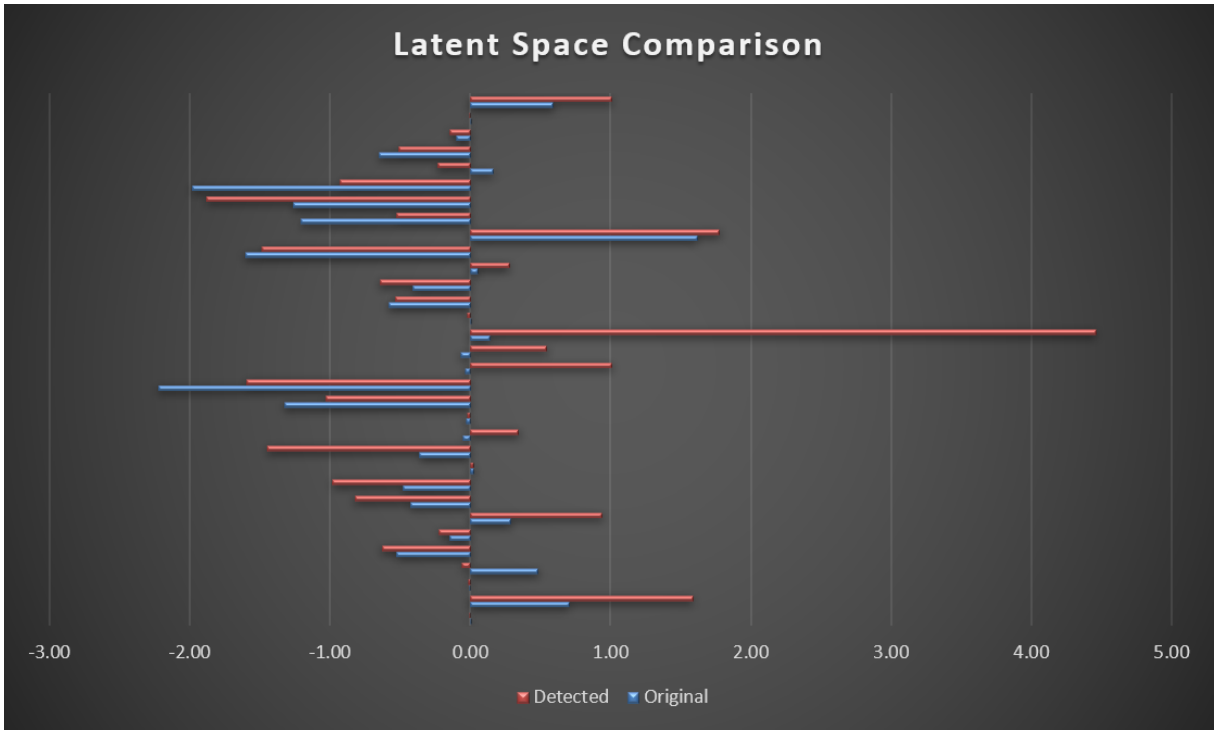

Fig. 14 Comparison of latent space.

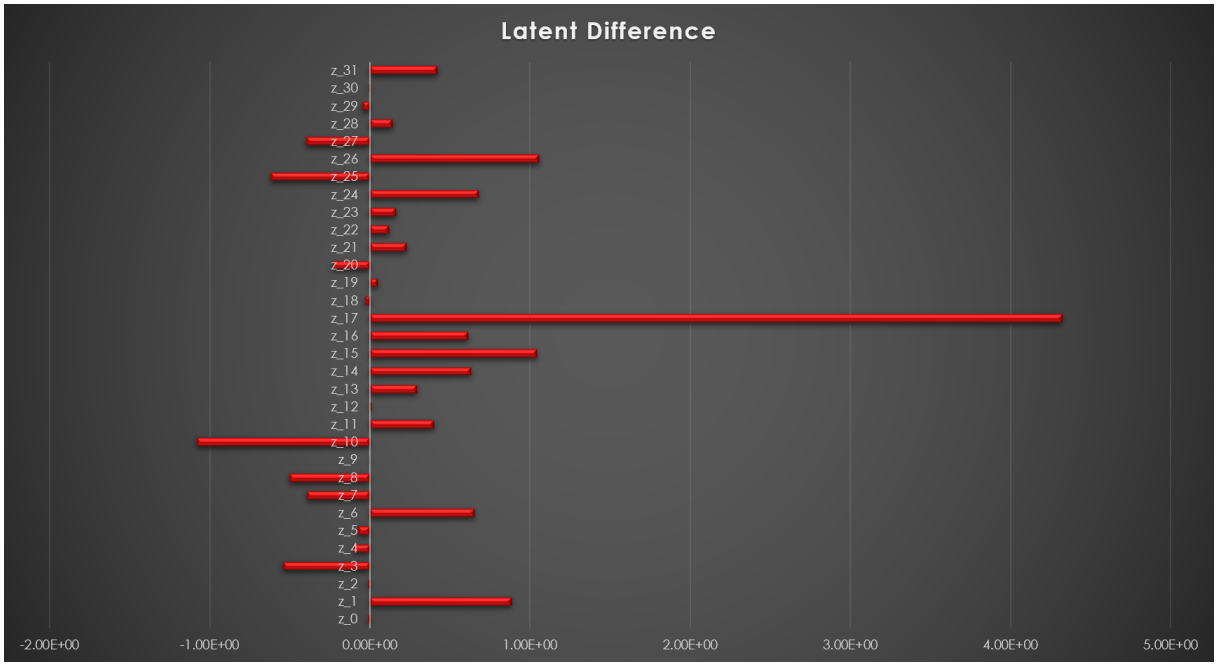

Fig. 15 Difference of latent space.

V. Conclusions

Many AI architectures such as neural networks are black boxes. Explaining intermediate results of the neural network to humans could increase the justifiable trust in the AI decisions. The goal of this effort was to enhance explainability of image-based neural networks with a focus on object classification. It was found that one of the difficulties of explaining a pretrained neural network is that the features are entangled together. For instance, a neuron that is highly selective to detecting long bird beaks could also be selective in detecting dogs. However, this work has found it feasible to induce disentanglement and explainability of neural networks if additional constraints are introduced. The β-TCVAE places a weight on neurons to more closely model Gaussian distributions in its latent space which causes interesting disentanglement of the data set. These more explainable neurons were shown to be useful in a classification task and these features could provide a human-level justification of decisions. This approach is early in development and is not yet applicable to a critical system. While some disentanglement is possible, a large of amount variation of a complex data set such as human faces is still entangled.

VI. Acknowledgments

This effort was supported by the ATTRACTOR subproject under the Convergent Aeronautics Solutions project of NASA ARMD's Transformative Aeronautics Concepts Program (TACP). The ATTRACTOR project was led by Dr. B. Danette Allen and Dr. Natalia Alexandrov.

References

[1] Gunning, D., "DARPA's Explainable Artificial Intelligence (XAI) Program," *Proceedings of the 24th International Conference on Intelligent User Interfaces*, Association for Computing Machinery, New York, NY, USA, 2019, p. ii. https://doi.org/10.1145/3301275.3308446, URL https://doi.org/10.1145/3301275.3308446.

[2] Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., and Mordvintsev, A., "The Building Blocks of Interpretability," *Distill*, 2018. https://doi.org/10.23915/distill.00010, https://distill.pub/2018/building-blocks.

[3] Guan, C., Wang, X., Zhang, Q., Chen, R., He, D., and Xie, X., "Towards a Deep and Unified Understanding of Deep Neural Models in NLP," PMLR, Long Beach, California, USA, 2019, pp. 2454–2463. URL http://proceedings.mlr.press/v97/guan19a.html.

[4] Ancona, M., Oztireli, C., and Gross, M., "Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Value Approximation," PMLR, Long Beach, California, USA, 2019, pp. 272–281. URL http://proceedings.mlr.press/v97/ancona19a.html.

[5] Guo, T., Lin, T., and Antulov-Fantulin, N., "Exploring interpretable LSTM neural networks over multi-variable data," PMLR, Long Beach, California, USA, 2019, pp. 2494–2504. URL http://proceedings.mlr.press/v97/guo19b.html.

[6] Etmann, C., Lunz, S., Maass, P., and Schoenlieb, C., "On the Connection Between Adversarial Robustness and Saliency Map Interpretability," PMLR, Long Beach, California, USA, 2019, pp. 1823–1832. URL http://proceedings.mlr.press/v97/etmann19a.html.

[7] Dolph, C. V., Tran, L., and Allen, B. D., *Towards Explainability of UAV-Based Convolutional Neural Networks for Object Classification*, ???? https://doi.org/10.2514/6.2018-4011, URL https://arc.aiaa.org/doi/abs/10.2514/6.2018-4011.

[8] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A., "Going Deeper with Convolutions," , 2014.

[9] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., "ImageNet: A Large-Scale Hierarchical Image Database," *CVPR09*, 2009.

[10] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M. M., Mohamed, S., and Lerchner, A., "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," *ICLR*, 2017.

[11] Rumelhart, D. E., and McClelland, J. L., *Learning Internal Representations by Error Propagation*, MITP, 1987, pp. 318–362. URL https://ieeexplore.ieee.org/document/6302929.

[12] Kingma, D. P., and Welling, M., "Auto-Encoding Variational Bayes," , 2013.

[13] Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A., "Understanding disentangling in β-VAE," , 2018.

[14] Chen, R. T. Q., Li, X., Grosse, R., and Duvenaud, D., "Isolating Sources of Disentanglement in Variational Autoencoders," , 2018.

[15] Li, Y., Pan, Q., Wang, S., Peng, H., Yang, T., and Cambria, E., "Disentangled Variational Auto-Encoder for Semi-supervised Learning," , 2017.

[16] Liu, Z., Luo, P., Wang, X., and Tang, X., "Deep Learning Face Attributes in the Wild," *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[17] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P., "Image Quality Assessment: From Error Visibility to Structural Similarity," *Trans. Img. Proc.*, Vol. 13, No. 4, 2004, p. 600–612. https://doi.org/10.1109/TIP.2003.819861, URL https://doi.org/10.1109/TIP.2003.819861.

Fig. 16 β-TCVAE latent traversal of azimuth rotation.

Fig. 17 Entangled β-TCVAE latent traversal of azimuth rotation and eyebrow height.

Fig. 18 β-TCVAE latent traversal of face rotation.

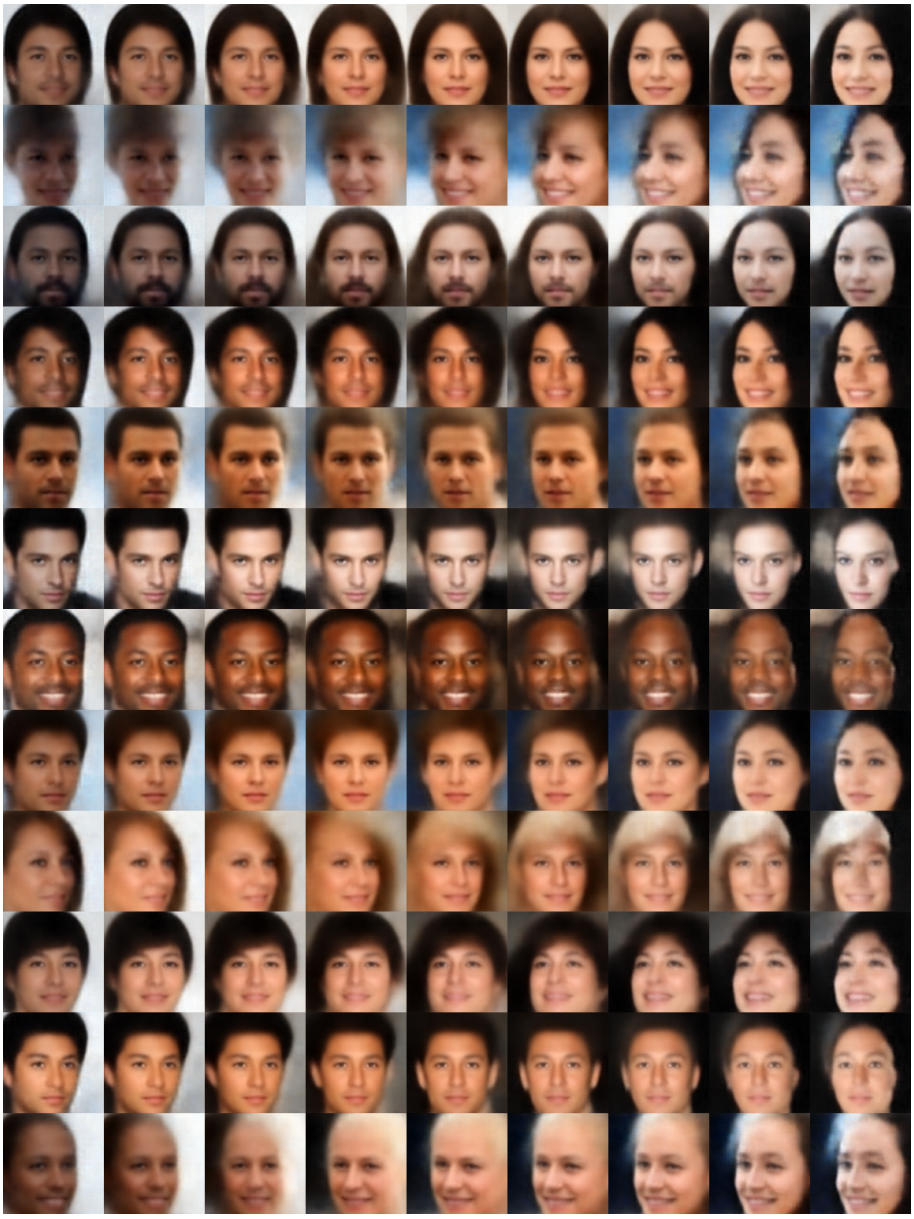

Fig. 19 Entangled β-TCVAE latent traversal of rotation, face shape, gender, shadow.

Fig. 20 β-TCVAE latent traversal of hair part.

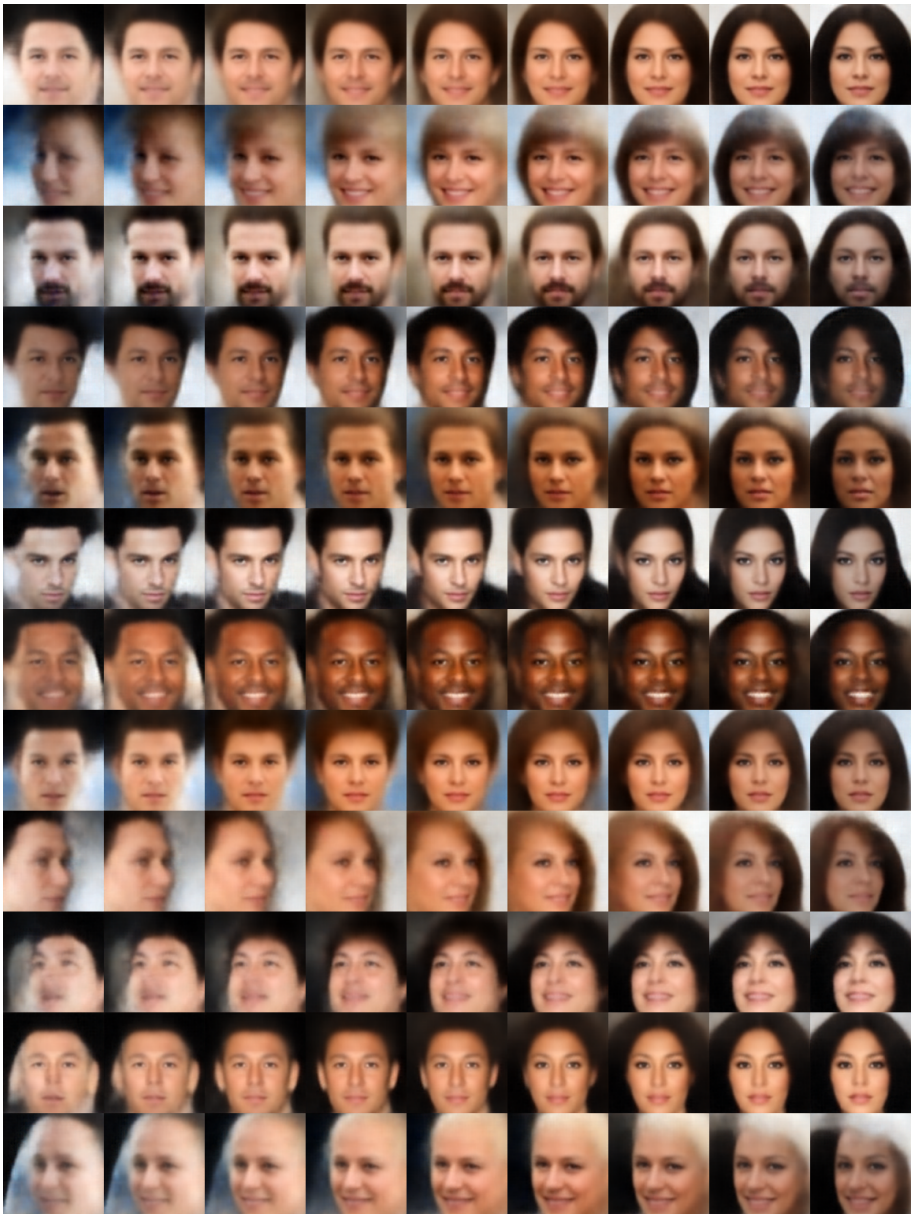

Fig. 21 β-TCVAE latent traversal of hair length.

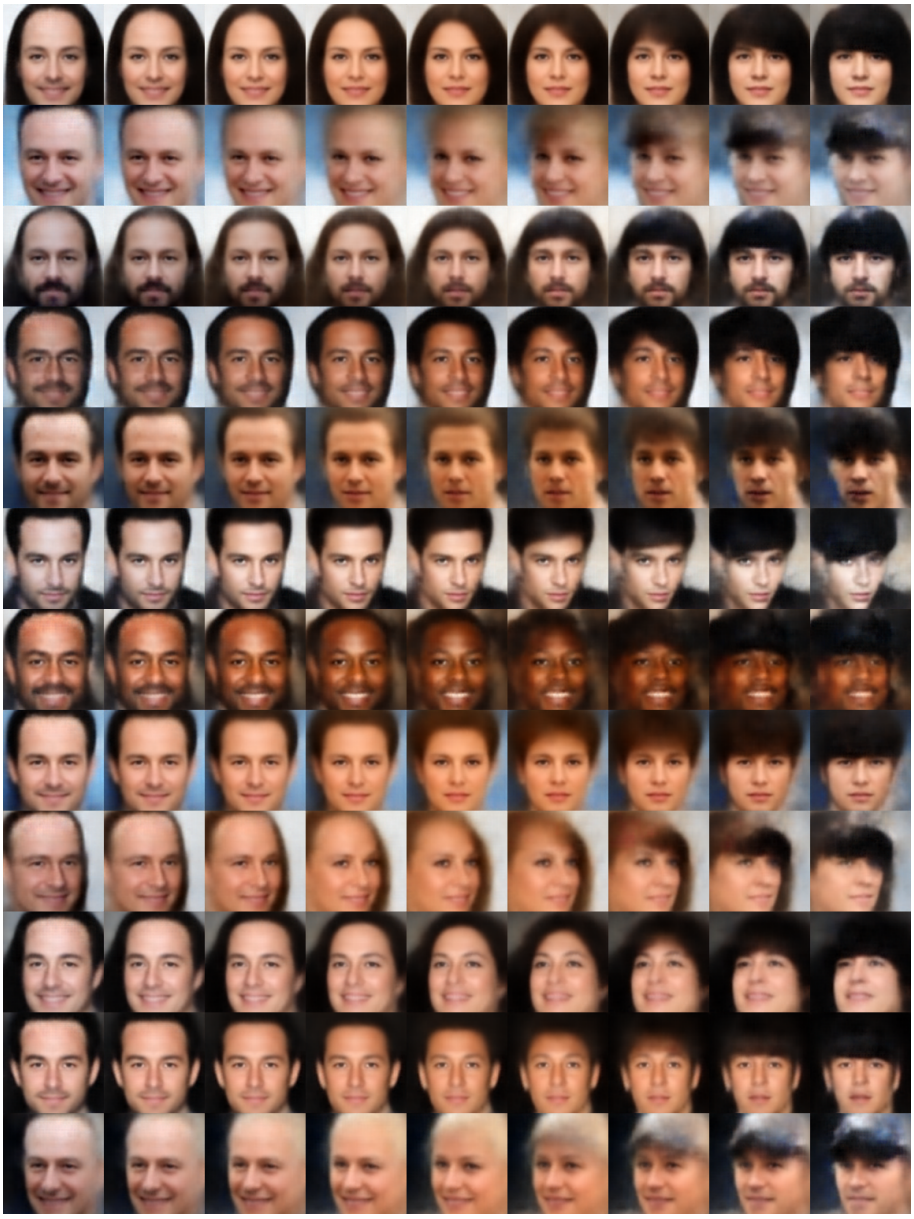

Fig. 22 β-TCVAE latent traversal of hairline.

Fig. 23 β-TCVAE latent traversal of shadow.

Fig. 24 β-TCVAE latent traversal of background color.

Fig. 25 β-TCVAE latent traversal of skin color.

Fig. 26 β-TCVAE latent traversal of sunglasses.

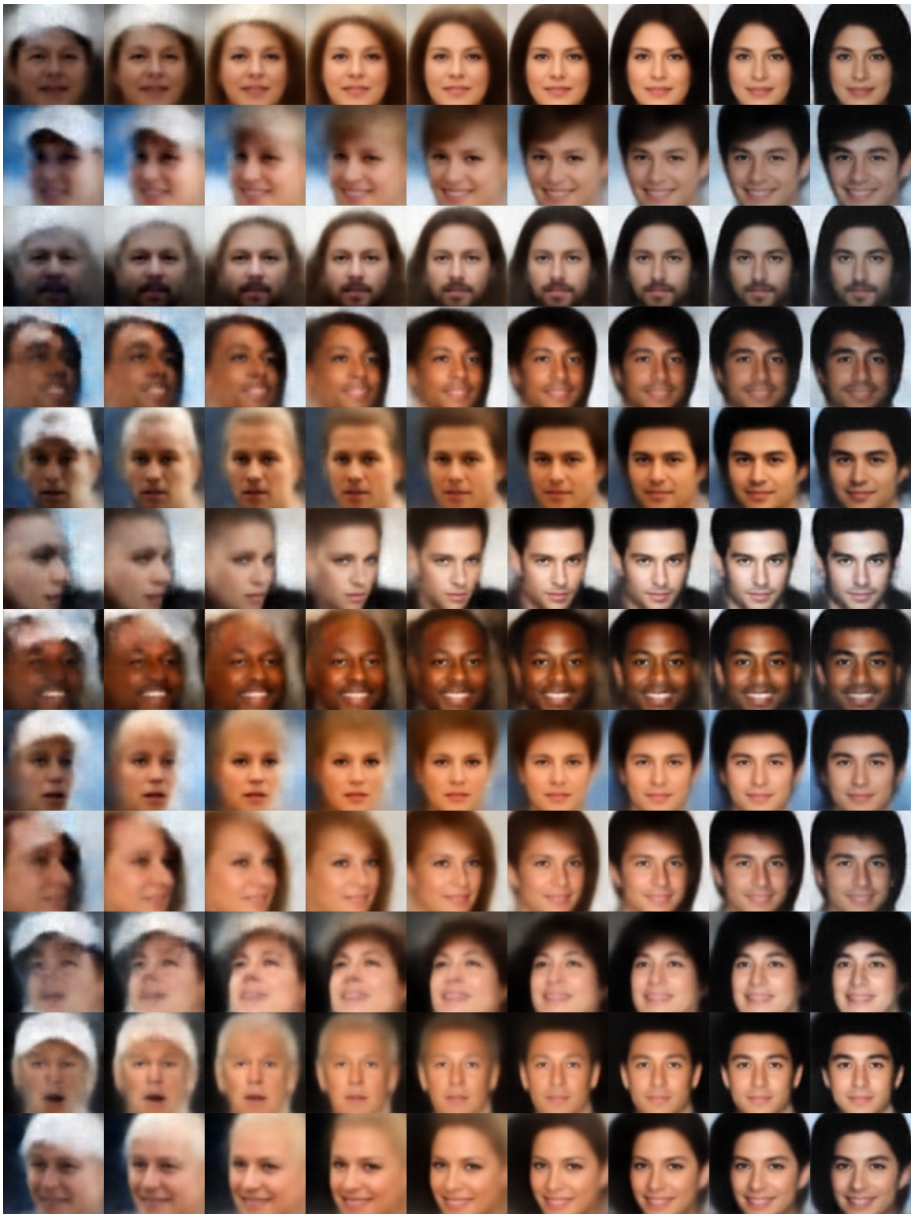

Fig. 27 Entangled β-TCVAE latent traversal of white hat and face detail.

Fig. 28 Entangled β-TCVAE latent traversal of smile and facial hair.

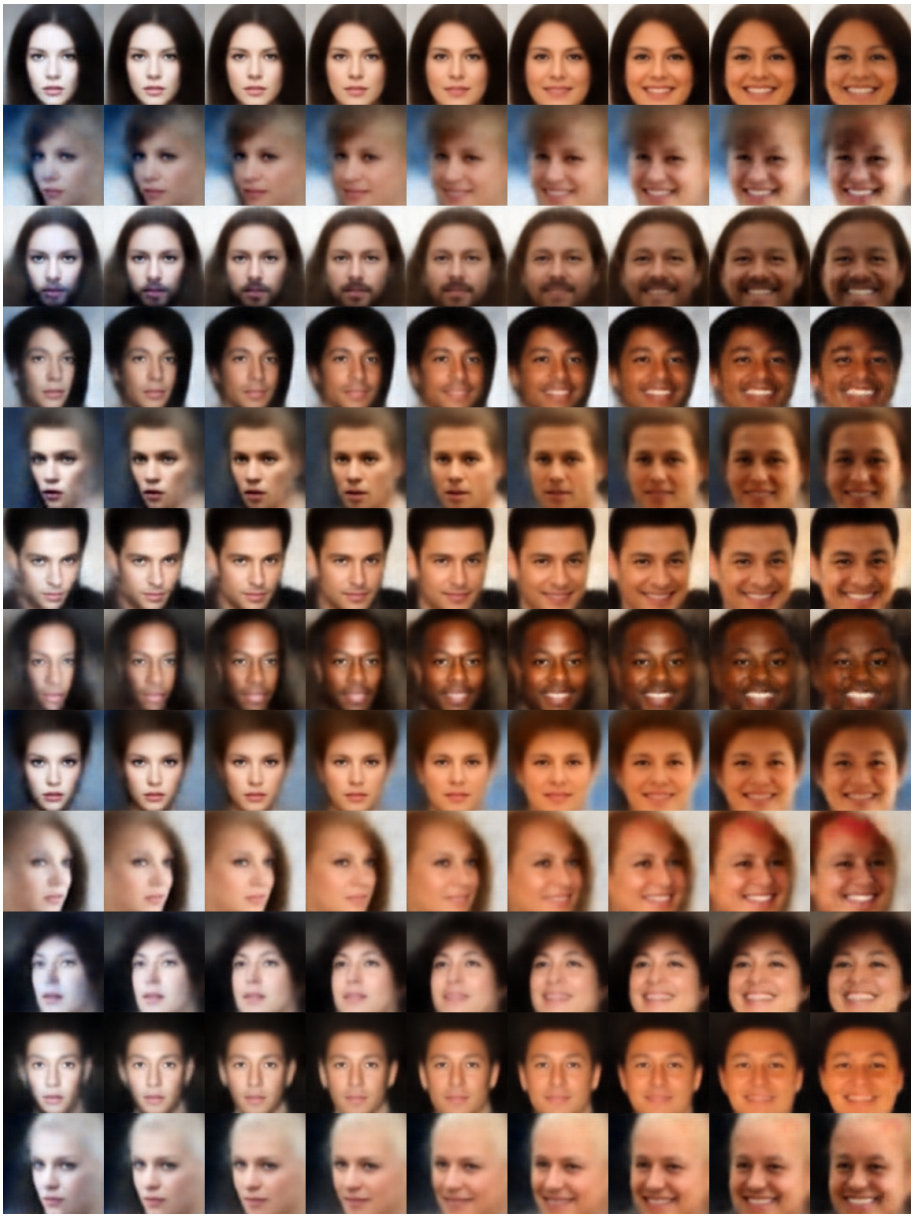

Fig. 29 Entangled β-TCVAE latent traversal of face width, skin color, and smile.