Enhancing Neural Network Decision-Making with Variational Autoencoders

Loc Tran and Derek Zhao and Chester Dolph

June 2, 2020

Abstract

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. This explainability encourages people to be more inclined to justifiably trust machine decision-making.

1 Introduction

An area of recent research interest is greater understanding of the internal mechanisms within neural networks [1]. The process of how deep neural networks make decisions is not fully understood. The meaning of the internal components and why image features are chosen in the training process for a given layer is not clear. This concept of understanding the internal workings is called neural network explainability or interpretability. 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 [2]. Feature visualization is a method of producing a visual representation of a network at the neuron, channel, or layer level. The visualization may be a manifestation of weights,

convolution filters, activation, gradients, neurons, the response to a given input image, an amplification of neural activity, reconstruction of the input from the response, or a combination of the aforementioned feature visualization techniques.

Our contributions include showing that it is feasible to generate explainable features from a neural network 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.

2 Description of the Problem

Currently, a CNN is often used as a black box where users train the network using an image data set, supply an input image, and the output is a decision with little to no supporting evidence to why the decision is made. A deep CNN is composed of multiple layers between the input and output. Researchers have been tuning CNNs to get improved classification accuracy and speed for their application while forgoing explainability of the system. The goal of this work is to achieve a greater understanding and explainability of CNN. Reducing the "black box" implementation of image classification via explainability will benefit the effective teaming of humans and machines where establishing trust between agents executing a shared mission.

Despite the success of CNN and their improved architectures, the internal mechanisms of what causes the high classification accuracy remain unclear. Substantial interest has been given to improving the classification accuracy of neural networks without focusing on the explainability of the inner layers. CNNs are commonly being used as black boxes where a structure is defined and then trained over large data sets. The emphasis placed on the resulting model is typically focused on achieving high test accuracy while forgoing transparency of the algorithm. Neural network interpretability is a burgeoning area of research with a few different approaches through feature visualization to better understand the inner workings of the neural networks. The imagery generated during feature visualization provides a means to provide human interpretability for neural networks. A better understanding of the internal networks may lead to: 1) Improved trust of CNN to perform safety critical tasks 2) Opportunity of retraining the network during a mission as new information is gained such as objects of interest 3) Improvement in CNN design by understanding which features and layers are most important for classification and which are superfluous.

3 Approach

Our approach was to first see if there was selectivity of the neurons in a pre-trained neural network [2]. A popular image classification neural network is GoogLeNet [3] which had state of the art classification performance on the ImageNet data set [4] 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?

We later explored ways to induce selectivity by training a 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 variational autoencoder [7]. 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.

4 Solution

4.1 Variational Autoencoders

An autoencoder [5] 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 estimate 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 propagating neural network latest into image space

helped in interpretation of the layer. Similar to conventional neural networks, slight modification of values in latent space are not currently explainable and could correspond to large deviations in the generated image.

The variational autoencoder (VAE) was introduced by Kingma et al [6]. 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 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.

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

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. All nodes in the latent space except for one. The effect of that particular latent feature could be isolated and viewed in image space.

Higgens et al introduce the β-VAE with a simple modification to the cost function [7]. A single weighting parameter is

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

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 [8], improve the reconstruction accuracy [9], and applying the network to semi supervised learning [10].

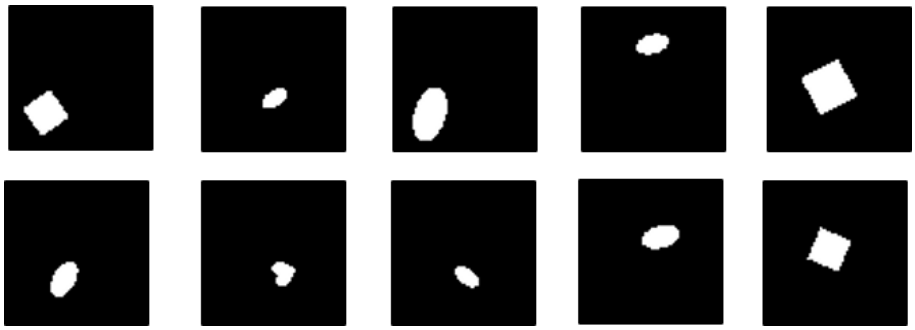

Figure 1: DSPRITES example image

5 Results

The first experiment on variational autoencoders was on the dSprites data set [11]. A sample of the images in this data set 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 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

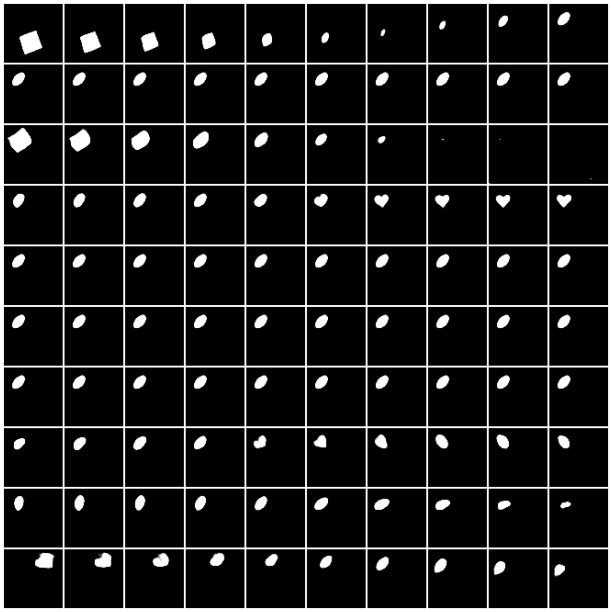

Figure 2: dSprites VAE traversal.

latent space.

The second data set used to analyze the VAE was CelebA [12]. 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 15 - Fig **??**. 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 15 shows bottom to top rotation, Fig 17 shows left to right rotation. Several the disentangled features related to hair variations such as Fig **??** which is selective to the location of the hair part, Fig **??** is selective to the hair length, and Fig **??** 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 **??** shows a latent feature that

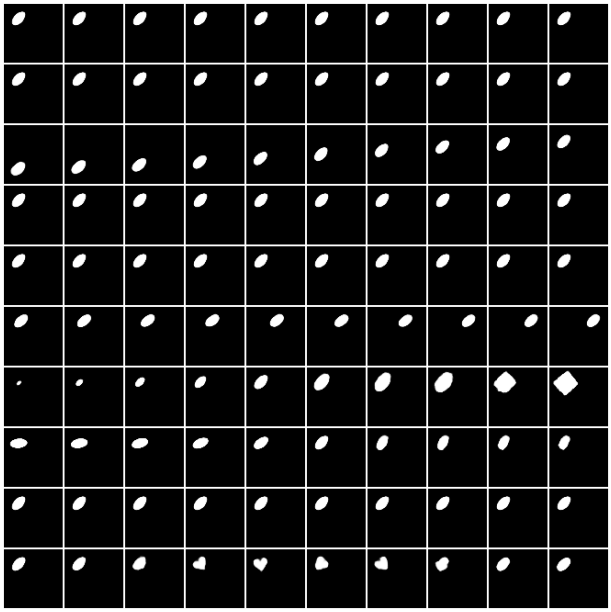

Figure 3: dSprites β-VAE traversal.

controlled the level of smile but was entangled with skin tone and facial hear. Similarly Fig **??** 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.

5.1 Analytical

The introduction of the β hyperpameter as a scaling factor of the KL divergence presents a balancing act. There are two opposing 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. As a result, opposing behavior occurs. 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.

An experiment was conducted to see if the disentangled features from a VAE could be used in a machine learning problem. A simple 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 VAE was trained on faces from the

CelebA data set with 1000 faces left out of the training set for testing. Also, 100 non-face images were extracted from the COCO data set to be added to the test set. This classification problem could also be seen as an anomaly detection problem.

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. For this experiment, six reconstruction loss functions were considered including mean square error (MSE), mean absolute error (MAE), cosine distance, structural similarity metric, cross entropy, and color entropy.

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 4. Precision and recall is defined as below:

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

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

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 or it 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.

While the average precision of this classifier is fairly low, note that the number of non-face images was also very low 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 5. It can be seen that mean square error and structural similarity metric have the largest coefficients and are more important to the classification compared to the other reconstruction losses.

Some interesting question that arise are, what images cause the classifier to fail

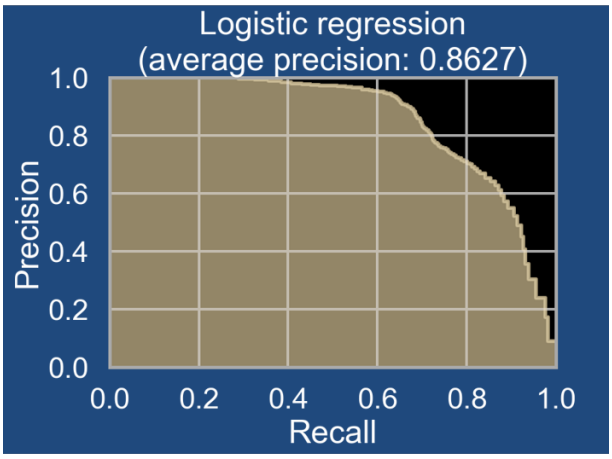

Figure 4: Precision-Recall curve.

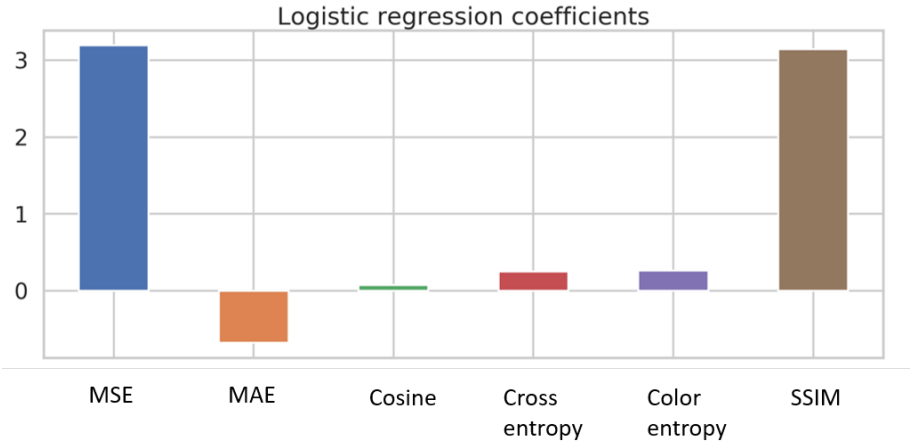

Figure 5: Precision-Recall curve.

Figure 6: Non-face images with the lowest mean square error.

Figure 7: Non-face images with the highest mean square error.

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 6, 7, 8, and 9 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 6 are that the images tend to not have a lot of detail. There is generally a large portion of the the input image that have a 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 8. 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 7. 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

Figure 8: Face images with the lowest mean square error.

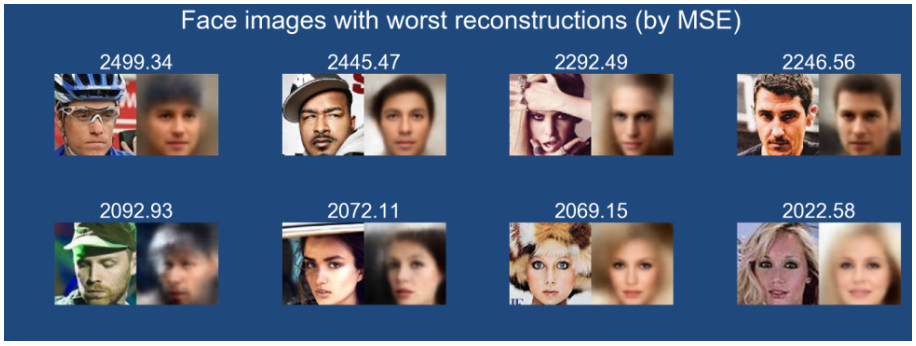

Figure 9: Face images with the highest mean square error.

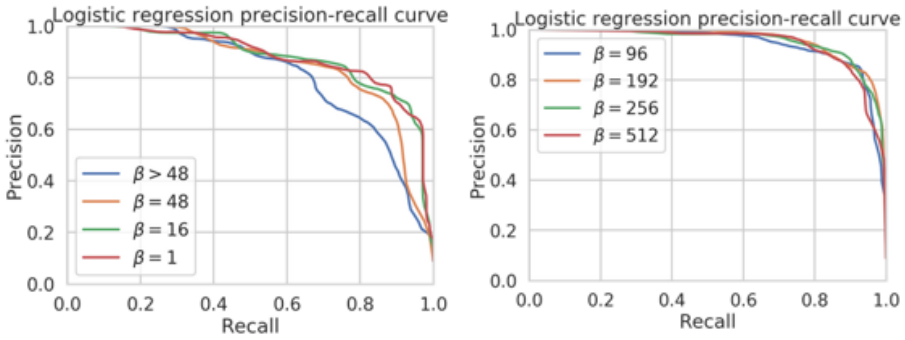

Figure 10: Precision recall curve using varying levels of β (Left) Using cosine distance loss function. (Right) Using SSIM reconstruction loss function.

similarity metric was selected since it more closely aligns with how a person judges the similarity of two images. The VAE was retrained using SSIM during the training phase and the experiment was repeated. Fig 10 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.

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 methods to add robustness. 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 11. 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.

5.2 Computational

The goal of this work was to enhance explainability of neural networks and no computational comparisons were generated.

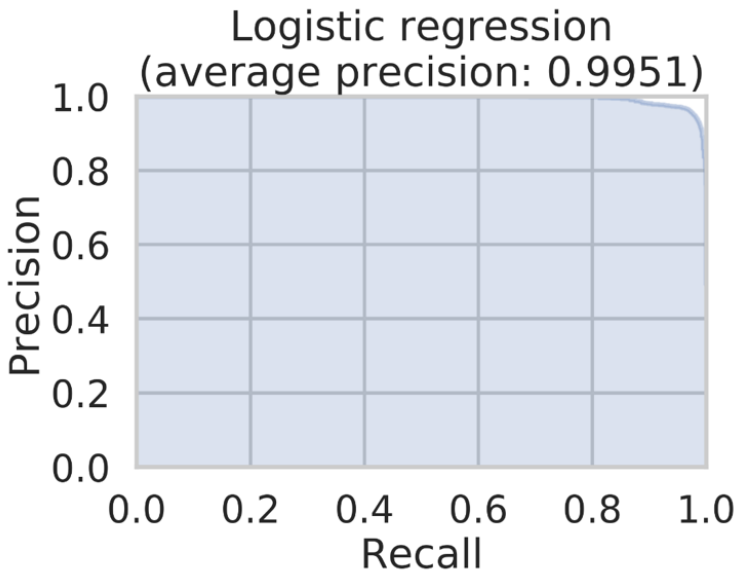

Figure 11: Precision-Recall curve using 1000 non-faces for training

Figure 12: Simulated person for search and rescue application.

5.3 Experimental

One of the disentangled features that was extracted by the beta VAE was the presence of sunglasses which is shown in Fig **??**. 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 the Baseline Environment for Autonomous Modeling (BEAM) simulation environment [13] to generate the sample images which are shown in Fig 12.

The VAE was applied to both the original image and the target image with the sunglasses. Fig 13 shows the latent encoding of the two images. The blue

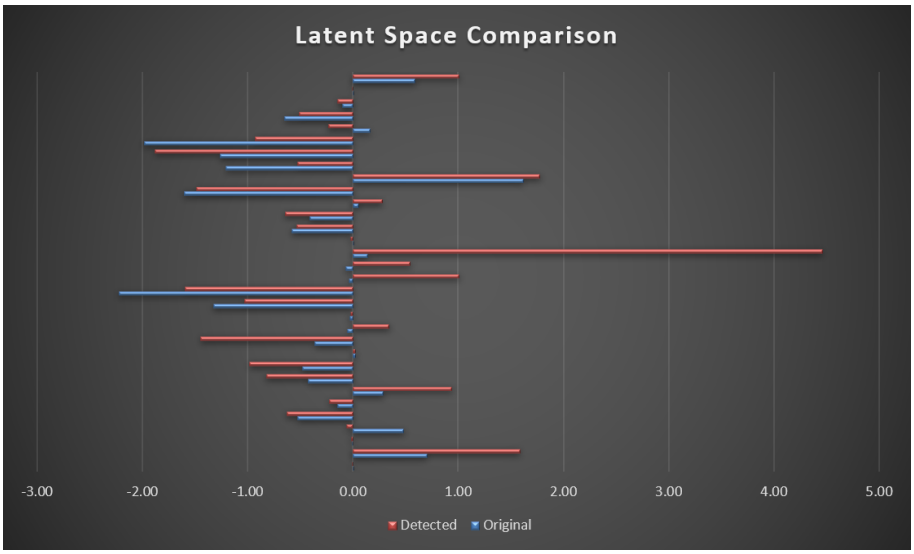

Figure 13: Comparison of latent space.

bars correspond to the original image while the red was from the target image.

To see the comparison better, Fig 14 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. In the exemplar case of a search and rescue mission. The VAE would 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.

6 Integration Status

The variational autoencoder work has been integrated into Langley's Autonomous Entity Operations Network (AEON) software architecture and Baseline Environment for Autonomous Modeling (BEAM) simulation environment [13]. The simulated tests in Section 5.3 required integration to BEAM which was also developed under ATTRACTOR. The source code for it has also been distributed to the Robust Software Engineering Group at Ames Research Center.

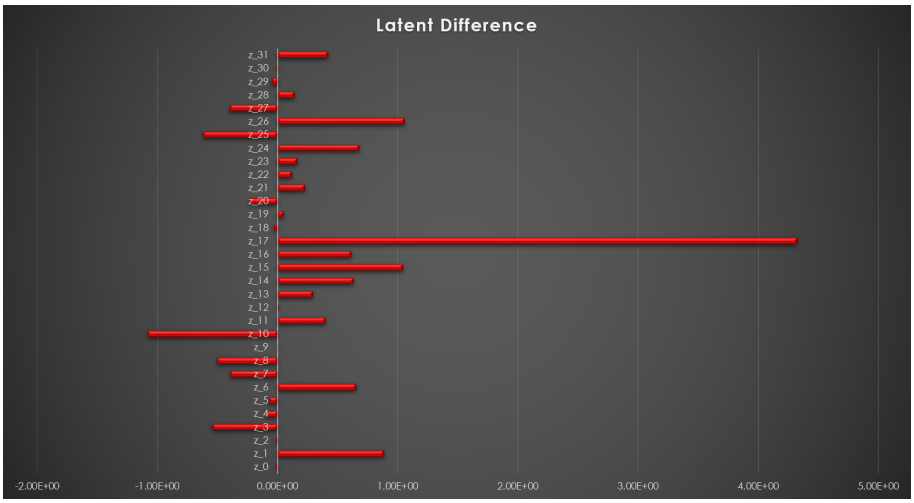

Figure 14: Difference of latent space.

7 Feasibility Conclusions

Many AI architectures such as neural networks are black boxes. Explaining intermediate results of the neural network to humans would 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 was found to 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-VAE 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 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.

8 Future Work

Currently, partial disentanglement of data sets can be achieved. Further improvements in disentangement could enhance explainability. Some research activities that could be pursued are different probability distributions in the

latent space. An interesting idea would be to see if the VAE can extract a feature with a known probability distribution function that is non-Gaussian.

References

[1] David Gunning. Darpa's explainable artificial intelligence (xai) program. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page ii, New York, NY, USA, 2019. Association for Computing Machinery.

[2] Chester V. Dolph, Loc Tran, and Bonnie D. Allen. *Towards Explainability of UAV-Based Convolutional Neural Networks for Object Classification.*

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.

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

[5] D. E. Rumelhart and J. L. McClelland. *Learning Internal Representations by Error Propagation*, pages 318–362. MITP, 1987.

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

[7] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew M Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[8] Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae, 2018.

[9] Ricky T. Q. Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders, 2018.

[10] Yang Li, Quan Pan, Suhang Wang, Haiyun Peng, Tao Yang, and Erik Cambria. Disentangled variational auto-encoder for semi-supervised learning, 2017.

[11] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[12] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[13] Benjamin Kelley, Ralph Williams, Jason Holland, Otto Schnarr, and B. Allen. A persistent simulation environment for autonomous systems. 06 2018.

Figure 15: β-VAE latent traversal of azimuth rotation.

Figure 16: Entangled β-VAE latent traversal of azimuth rotation and eyebrow height.

Figure 17: β-VAE latent traversal of face rotation.

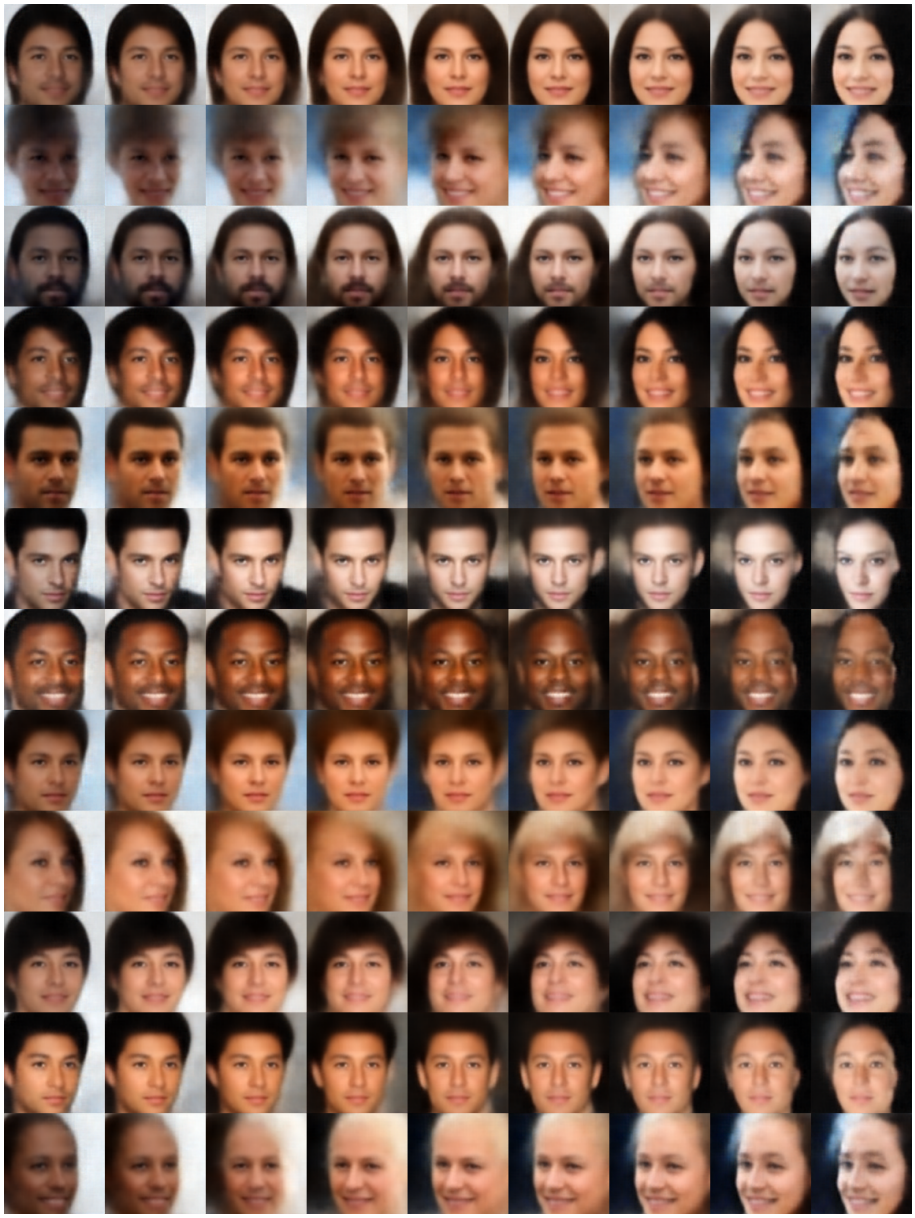

Figure 18: Entangled β-VAE latent traversal of rotation, face shape, gender, shadow.