Interpretation of Activation Maps in Generative Models

Prabhakar Kumar Panday[1] **Praveen Kumar**[1] **Srijay Kolevkar**[1]

Abstract

Recent work in the field of Explainable AI and Computer Vision on CNN based architecture has improved the interpretability of Deep Learning models and helped in visualizing the model predictions. Methods like CAM, Grad-CAM and Guided Grad-CAM have proved the practicality of localized visual attention in the classification and categorization applications. However, not much research has been done on generative models. In our work, we implement Grad-CAM technique on VAE and CVAE models trained on CelebA-HQ dataset and calculate neural attention map. The aim of the project is to build generative models capable of generating controllable human faces and build semantic segmentation of human face, and then investigate methodologies to improve the explainability by applying Explainable AI techniques like Grad-CAM, and analysing the effect of altering the model architecture, loss functions, latent space size. Furthermore, we investigate the latent space information of models by modifying the latent node variables.

1. Introduction

Deep learning (DL) is introduced in various domains and applications due to its enormous potential of improving the existing systems. Many deep learning applications have been demonstrated in the field of Natural Language Processing (Liddy, 2001), Computer Vision (Szeliski, 2010) (Parker, 2010), Robotics and Reinforcement learning (Kaelbling et al., 1996) etc. with a wide spectrum of the applications.

Deep Learning can broadly be divided into 2 major branches, namely supervised learning and unsupervised learning. Supervised learning tasks performs mapping of input to an output. For each input from the dataset, there is a corresponding desired output label or ground truth, which to-gether represent the training data. The model uses ground truth for training the Neural Network (NN) model and try to learn the underlying features or attributes of the dataset. Supervised learning is predominantly used to solve problems related to classification and regression, and it needs the dataset to be labelled or annotated, which is often not readily available. Unsupervised learning is a branch of DL in which data without label attribute is used for training. The model automatically learns hidden patterns in the data by extracting features with the help of methods like clustering (Caron et al., 2018), dimension reduction (Fodor, 2002), and representation learning (Bengio et al., 2013). Generative modelling is a type of unsupervised learning, which learns the underlying distribution in the training data and is capable of generating new data points. In this process of learning, the model is forced to imbibe the intricate and concise representation of training data. To give an intuitive insight, a supervised model can be trained to classify a painting and photograph accurately, but a generative model can be trained to create a painting.

Variational Autoencoders (VAE) is a type of generative model based on the foundational work of variational Bayesian methods. VAE consist of two main parts namely, Encoder and Decoder. Encoder is responsible for reducing the dimensionality of input data and thus focusing on only the salient features of the input training data. On the contrary, Decoder reconstructs the output of Encoder to a pre-defined input data size. Encoder and Decoder of image data application based VAE models have convolution based layers(Gu et al., 2018). Output of these layers are called activation maps. The intermediate output between Encoder and Decoder is called latent space. Architecture of VAE model will be further explained in detail in next sections.

In general, Deep Neural Network (DNN) models perform good on well-defined tasks, however, they are not transparent in the operation and behave in an opaque manner (Rai, 2020). Thus, DNN models are considered as black boxes with no strong intuition behind the working and predictions of the model. In Computer Vision domain and its consumer focusing applications like healthcare (Holzinger et al., 2017) and self-driving technology (Grzywaczewski, 2017), there is a strong need for the reasoning behind the working of the model to build trust and reliability on the DNN model before deployment. Lack of clear understanding can lead to

---

[1]Universität Stuttgart, Germany. Correspondence to: Prabhakar Kumar Panday, Praveen Kumar, Srijay Kolvekar <st169844,st170707,st169713@stud.uni-stuttgart.de>.

adverse consequences.

In this work, we focus on generative models, more specifically VAE and its variations. Moreover, we try to proffer an interpretation of the activation map outputs from the convolutional layers of Encoder and the intermediate latent space of generative model.

This paper is organized into six major parts. In the first section, we give the introduction, which further includes the motivation and research ground for the work. The second part of this document gives the necessary theoretical background to understand this work. Here, we describe basic working of NN models and also cover different types of generative models used in this work. The third section outlines the relevant previous work that has been done on interpretation and explanation of generative models. In the fourth section, we give a thorough description of the methods used to give the interpretation of the activation maps of generative models. In the fifth and sixth section, we provide the results and conclusion of our work, along with a hint for the scope of future work.

2. Theory

This section gives a short introduction to relevant theoretical topics that are important to understand this work.

2.1. Deep Learning

Deep neural networks (DNN) are becoming the most popular solution for solving complex problems without performing explicit feature extraction. In the very basic form, a DNN consists of more than two hidden layers, and each layer consists of multiple neurons. Input data is passed through each layer to finally get a prediction at output. The important aspect of the neural network is that the output of each layer is passed through a non-linear activation function.

In the basic form of DNN, neurons of two dense neural network layers are fully connected to each other. The input data is flattened into a single long vector, which is further traversed through each layer. The output of a layer l is the output of the activation function of that layer, given as $x_l = \phi_l(W_l * x_l - 1 + b_l)$. Where, $x_l - 1$ is the input vector for layer l and has dimension N. b_l is the bias vector for layer l and has dimension M. W_l is the weight matrix for layer l and has dimension MN. $\phi_l()$ is a vector activation function for layer l. x_l is an output vector for layer l and has dimension M. There are different types of activation functions, the most popular ones are Rectifier Linear Unit (ReLU) and its variations like Leaky ReLU and Exponential linear unit (ELU).

In practice, more advanced techniques like Convolutional Neural Network(CNN) (Albawi et al., 2017), Recurrent

Neural Network (Zaremba et al., 2014), Generative Adversarial Network (Aggarwal et al., 2021) are used, also it is very common to use a combination of techniques to achieve desired results.

Convolutional Neural Network(CNN) layers are one of the most popular layers used in tasks related to visual imagery. They encode spatial information into the subsequent layers by performing convolutional operations. Filters, or commonly referred to as kernels, are used to slide above the image to capture the low and high level features of the input image as shown in figure 1. CNN are usually coupled with pooling layers, Pooling layers reduce the dimension of data from previous layers by performing average operation within a feature map. Global Average Pooling(GAP) is one such operation, where all neuron weights from previous layers are averaged to get a single value in the next layer.
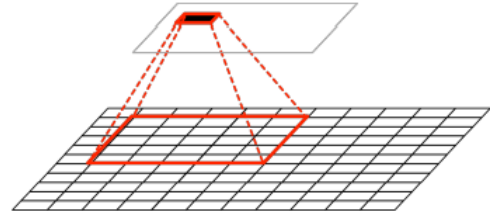

Figure 1. A kernel filter sliding over the input image (Gu et al., 2018)

2.2. Explainable AI (XAI)

Neural networks act as a black box, where most of the internal linear operations are either unclear or known with very little information. Explainable AI refers to techniques that help in unravelling this black box and understand why the neural network is performing the intended way. It aims to understand the reasoning behind the decisions of the neural network and most importantly, it aims to keep these explanations human understandable. Clear understanding of model's internal operations not only explains about the working of the model, but also provide intricate details which can be further used as additional information to improve the model efficiency.

2.3. Autoencoder

The Autoencoder(AE) is a special type of dimensionality reduction technique, which contains two major parts, namely Encoder and Decoder. The Encoder and Decoder contain multiple CNN layers along with some pooling layers. These CNN layers are capable of learning the low level and high-level features of image data. The Encoder is responsible to compress the input data x with unknown distribution $p(x)$ to a specific latent dimension z with distribution $p(z)$.

The latent dimension contains the mapped information of encoded input data. For instance, in figure 2, the latent dimension contains the compressed information on features of the image. In the second part Decoder, the same latent information z is used to reconstruct the image.
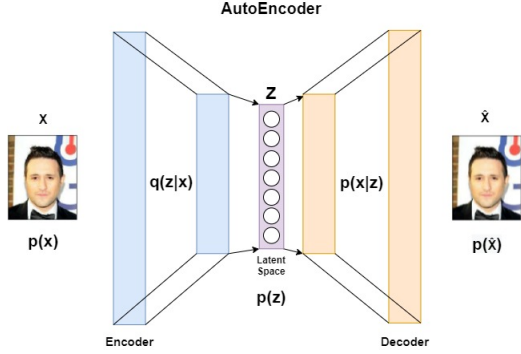

Figure 2. Architecture of a Autoencoder

2.4. Variational Autoencoder

Variational Autoencoder (VAE) (Kingma & Welling, 2019) is an unsupervised learning technique that works briefly on the architecture of Autoencoder, but with small variation. VAE is a generative model that uses a probabilistic approach in defining the latent space. It has an additional sampling approach, where instead of mapping a latent dimension to a vector, the latent dimension is mapped to a known Gaussian distribution $p(z)$ with a specified mean and variance, as shown in figure 3. In this way, each feature of the image is mapped to a certain normal distribution, which is represented by $p(z)$. The mapped latent distribution is further used by the Decoder to generate the new image $p(x|z)$ based on the distribution information.
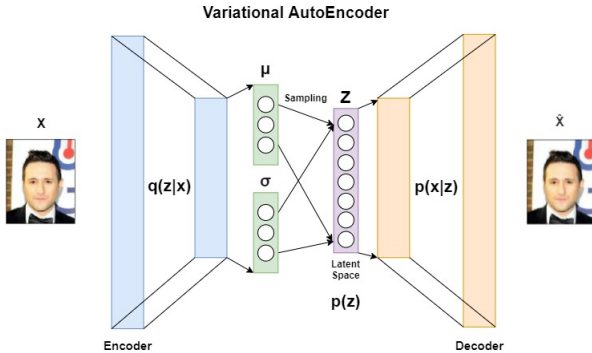

Figure 3. Architecture of a VAE

The widely used loss function for training the VAE are Kullback Leibler (KL) divergence and reconstruction loss (Kingma & Welling, 2019). Both losses are combined and used while training the model, as shown in equation 1.

$$E_{q(z|x)} \log p(x|z) - KL(q(z|x)||p(z)) \qquad (1)$$

where x is input data, z is a latent variable, $p(x)$ probability distribution of the input data, which is unknown. $q(z|x)$ defines the distinction of mapping x to latent variable z. $p(z)$ is the probability distribution of latent variable and $p(x|z)$ is the distribution of generated data given latent variable.

The KL divergence loss ensures that the learned Encoder distribution $q(z|x)$ maps to a well-defined Gaussian distribution $p(z)$. The KL divergence finds how two given distributions are different from each other. Hence, with this loss the Encoder tries to map the unknown distribution $p(x)$ to know $p(z)$. The reconstruction loss calculates maximum likelihood estimation and is specific to the Decoder. Reconstruction loss helps the Decoder reconstruct the image from the given distribution $p(z)$.

For the model to learn and find the global minimum, we use backpropagation(LeCun et al., 1989) algorithm. However, with VAE, the first element in the above-mentioned loss function (Equation 1) is stochastic variable z. Hence, backpropagation is not possible as we cannot take derivative. To solve this problem, we use the reparametrization trick(Kingma et al., 2015), where the stochastic z is converted to a deterministic variable by adding ϵ as described in the equation 2, where $\epsilon \in \mathcal{N}(0, 1)$.

$$z \in \mathcal{N}(\mu, \sigma^2) \rightarrow z = \mu + \sigma \qquad (2)$$

2.5. Conditional Variational Autoencoder

Conditional VAE (CVAE) is similar to the above-mentioned VAE model in terms of architecture and loss function. However, in the case of CVAE, we use additional information in the form of condition C. The condition contains information of the features of the input data. For example, in the case of input image data, the image contains certain feature or attributes like colour, smile, hair etc. These attributes define an image. In CVAE, along with the input image, the condition or the image attributes are fed to the network which is in turn mapped to the Gaussian distribution $p(z|c)$. Hence, the neural architecture has additional information of conditional attributes which is then used by the Encoder to map the low-level and high-level features.

The special feature of the CVAE is that given the noise data along with the conditional features, the model is now able to generate the new image which contains the mentioned conditional features.
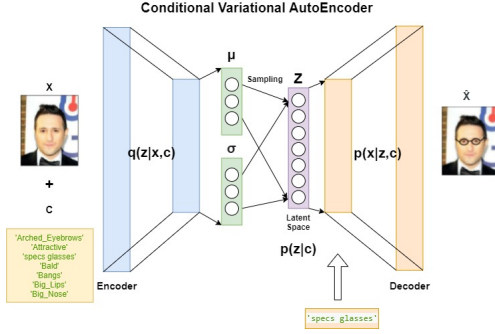
Figure 4. Architecture of a CVAE

3. Related Works

This section describes previous work done in explaining neural network models.

CNN based architectures are used in variety of applications and domains due to its capabilities in image based operations. However, these model lack the interpretability (Chakraborty et al., 2017). To tackle this issue related to the interpretability, many methods have been described. Zhou et al. 2016 proposed the technique of Class Activation Mapping (CAM) for CNNs with Global Average Pooling(GAP), which accurately perform object localization. Nevertheless, the approach mentioned in the work requires the specific architecture type, hence cannot be generalized. Zhou et al. 2016, Zeiler & Fergus 2014, Selvaraju et al. 2017 visualize the activity within the CNN model's hidden inner layers, such as compositionality and class discrimination and expose properties used to build saliency maps. One of these methods, Grad Class Activation Map or Grad-CAM was introduced by Selvaraju et al. 2017 as an Explainable AI technique. It helps visualize which input features were most relevant in making a certain decision by the neural network using heatmaps. A heatmap is a representation in which magnitudes of data points are represented in a 2-dimensional plane using gradually varying shades of colour. The shades may vary from blue to red, indicating an increasing significance. Moreover this approach of visualizing the activation map is generalized and can be applied to different architectures with very little modification.

Although many methods are described to explain the CNN based discriminative models, very less focus has been shed towards the visual explanation of generative model. The work by Liu et al. 2020 describes a method to visually explain the generative model. Our work mainly takes into account the method described in the (Liu et al., 2020) to use Grad CAM to visualize latent space information. This method will be further explored in next section (4).

Another method for semantic segmentation is by Vinogradova et al. 2020, which implements Grad-CAM tech-

nique onto the U-net (Ronneberger et al., 2015) by training the model for semantic segmentation on Cityscapes dataset (Cordts et al., 2016). In the mentioned approach, the Grad-CAM technique was implemented on to the bottleneck (end of the Encoder before upsampling) layer of the U-net model.

4. Method

This section describes the methodologies implemented as part of this work. We capitalize on three methods to explain the activation maps of generative models. The first one is based on Grad-CAM (Liu et al., 2020), introduced in previous section (3), the second method manipulates the latent variables to visualize latent space encoding of VAE and CVAE. Third is semantic segmentation using VAE (Vinogradova et al., 2020).

4.1. Grad-CAM on VAE and CVAE

We first give an explanation on implementation of Grad CAM on CNN based classification model and then describe its adaptation to VAE and CVAE.

4.1.1. GRAD-CAM

Application of Grad-CAM to CNN based classification model is straight forward and requires no change to the original architecture or any retraining. Grad-CAM produces a heatmap that can be overlaid on top of the input image to clearly see which parts/features of the input image were most relevant in making the decision by the neutral network. Figure 5 shows an instance of Grad-CAM generated heatmap. As explained by Selvaraju et al. 2017, to calculate

Figure 5. Grad CAM generated heatmap overlaid(right) on the input image(left), highlighting parts of image most significant for prediction

Grad-CAM, three aspects of the model are required. First is the output class values before a softmax layer denoted by Y_c for c classes. Next, the activation map values are are the output of the last convolutional layer in the model's architecture, regarded as A_k, for each activation layer k.

Gradients are calculated as differentiation dY_c/dA_k of class outputs w.r.t activation maps as described in the equation 4. GAP is applied to the calculated gradients to get the value of α(equation 4). α is later multiplied with the activation maps to get attention maps(equation 3). Attention maps are oversampled to match the input image size and converted to gradient based colour map to get the final heatmap. Figure 6 gives an insight into the process flow of generating a Grad-CAM. i and j span the length and width of an activation layer k, Y represents the GAP operation applied on the gradient values for the entire activation layer.

$$M^c = ReLU(\sum_{k=1}^{n} \alpha_k A_k) \quad (3)$$

$$\alpha_k = \frac{1}{T} \sum_{p=1}^{h} \sum_{q=1}^{w} (\frac{\partial Y_c}{\partial A_k^{pq}}) \quad (4)$$
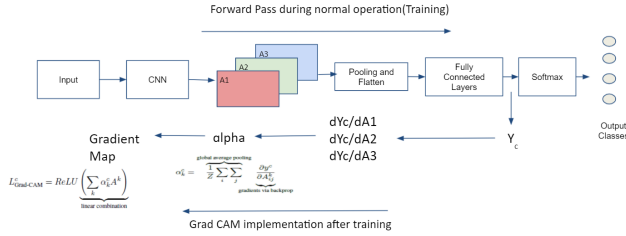

Figure 6. Implementation of Grad CAM on CNN based architecture

4.1.2. GRAD-CAM ON VAE AND CVAE

As explained in the section (2.4, 2.5), the VAE and CVAE architecture has an Encoder and the Decoder part. In CVAE, we have control over the data generation process using the conditions or the attributes of the dataset. In contrast, VAE does not have additional conditional inputs. In this part, we explain the implementation of VAE and CVAE architectures, and we also discuss integration of Grad-CAM on them.

The Encoder part contains eight Convolutional Neural Network (CNN) layers, with a max-pooling layer between two consecutive CNN layers. The kernel size ranges from 32 to 256 and the size of the actual image is compressed or downsampled. A flatten layer is used at the end of the last convolutional layer to push the compressed feature space to a vectorized array. Furthermore, sampling is carried out at this step to ensure the vectorized array is constrained to a Gaussian distribution with mean 0 and variance 1. This resulting vector is called latent space or bottleneck, which contains the compressed feature space information of the data set constrained in Gaussian distribution.

Similarly, the Decoder part of CVAE contains eight CNN layers with max-pooling layers but with the inverse operation of upsampling. The kernel size ranges from 256 to 32 with the image being upsampled to the original size. As mentioned in the section 2.4 and 2.5, the latent space information from a Gaussian distribution is used to generate a new image from the given distribution.The VAE and CVAE model is trained on the CelebA dataset5.4. Models are further trained with Adam optimizer. KL divergence and reconstruction loss functions are used to train the model, as mentioned in the section 3.

The application process of Grad-CAM has to be tweaked to suit the unique architecture of VAE and CVAE. With VAE, the intermediate latent space between Encoder and Decoder acts as class output Y_c. The output of the convolutional layer of the Encoder gives the activation maps for gradient calculation. The rest of the operation remains the same as for classification models(4.1.1).

4.2. Exploring the Latent Space

In this approach, we try to visualize the scope of information encoded in the latent space nodes. For this, we take fully trained VAE and CVAE models, and generate output images for a set of input images. Afterwards, we modify the value of individual latent space nodes gradually by adding values ranging from 0.1 to 0.9, and images are generated for each individual iteration. Furthermore, a heatmap is also generated using the Grad-CAM method for each iteration. The aim is to visualize the features encoded in the individual latent nodes and see the effect of modifying latent space node weights on output image.

4.3. Semantic Segmentation on VAE Models

In this section, we use VAE model for semantic segmentation of human faces from CelebA-HQ Dataset (5.4). Facial features like eyes, nose, lips, face structure are highlighted on the output. Once the VAE model is trained, Grad-CAM is implemented on the last layer of the Encoder of the VAE and the latent space. The loss metric for training was changed to Jaccarda loss and IOU score. As explained in the section 4.1.1, Grad-CAM generated heatmap is used to visualize the parts of the image which affect the output of the model as shown in figure 7. It can be clearly inferred that the model was able to recognize the details of human face and even the facial expressions of the person. The entire experiment is repeated with different VAE architectures, image sizes and latent dimensions to better interpret the model. The size of the image is varied from 64x64x3 to 256x256x3. Furthermore, latent dimensions are varied in the steps as follows: 5,15,50 and 100.

As shown in the table in figure 8, complicated models with smaller input image and smaller latent dimension was able
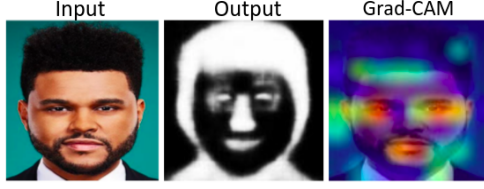
Figure 7. Input is a sample image from CelebA-HQ dataset(Left); Output face mask of the VAE model for a given image(Centre); Grad-CAM generated heatmap(Right)

Figure 9. Input image, image generated by VAE, Grad-CAM generated heatmap

to produce sharper masks than the models with large size images. On the other hand, models which take larger input images and has bigger latent space dimension produce less sharp images, but their interpretability is better. There is a sweet spot for input image size and latent dimension size, where the output image is clear and the model is less complex and interpretability is good.

for the reconstruction of the image is distributed across the activation maps.

Next, as per the method discussed in 4.2, we modified individual latent space nodes and generated the output images. The generated images and the Grad-CAM generated heatmap for that specific latent node are shown in figure 10.

Model Inputs/Outputs	64x64x3	128x128x3	256x256x3
CNN Layers	5	4	3
Latent Dimensions	5	15	100
Output Size	64x64x1	128x128x1	256x256x1
Sample Output			

Figure 8. Shows the output of various VAE model and the model architectures.

5. Results

In this section, we discuss the results of the implementation of Grad-CAM on VAE, CVAE models, and Semantic Segmentation on VAE models and then discuss how well it helps in explaining the activation maps of a generative model. Furthermore, we also assess if the method of modifying the latent space nodes, as discussed in section 4.2, gives any intuitive explanation about activation maps of VAE and CVAE.

5.1. VAE

At first, using the techniques discussed in section 4.1, we applied Grad-CAM on every latent space node and activation maps of the VAE model trained on CelebA dataset (5.4). These generated heatmaps are averaged to create a final heatmap, which is then overlaid on top of the input image, as shown in figure 9.

Clearly, this does not give a lot of useful insight, however, this does tell us that the importance of different features
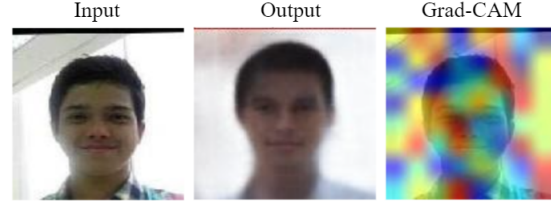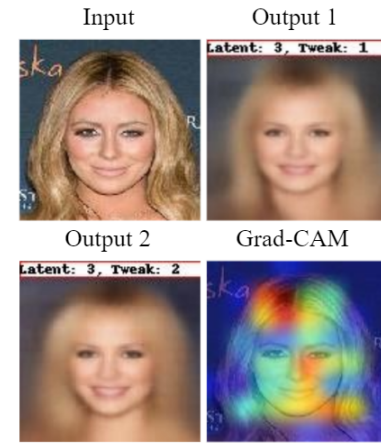

Figure 10. Input image(Top-left), Image generated by adding latent node 3 with value 0.1(Top-right), Image generated by adding latent node 3 with value 0.5(Bottom-left), Grad-CAM generated heatmap for latent node 3(Bottom-right)

As can be seen in the figure 10 that modifying the values of latent space node affects different features of the generated image including smile, hair pattern and the orientation of head pose.

5.2. CVAE

The CVAE based model showed considerable reconstruction capability as shown in the figure 11. However, the Grad-CAM results do not signify the exact reconstruction of image from the latent space, i.e. there were no specific features mapped onto a certain latent space which was clearly represented in the heatmap shown in the figure 11, 12.

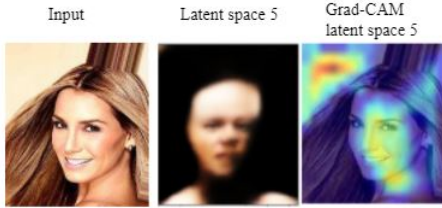*Figure 11.* CVAE reconstruction with Grad-CAM Implementation

Figure 12. CVAE Latent Space 5 with Grad-CAM Implementation

The latent dimension of each node represented multiple features of the image from different positions. This feature entanglement will lead to slight uneven gradient heatmap. The second important observation is related number of latent dimensions in the CVAE model. It is evident from the experimentation that the smaller number of latent dimension lead to entanglement of features in the latent space.

5.3. Semantic Segmentation on VAE Models

The semantic segmentation output of the VAE models are promising, the position of the facial features like eyes, nose, hair, face shape, lips, eyebrows were distinct. The segmentation model output, achieved a Jaccarda index loss of 0.5044 and IOU score of 0.76 for validation images.

The model was built with smaller sized images and latent dimensions respectively, and heatmaps were generated for each node of the latent space, using Grad-CAM described in section 4.1.2 . Next, we tweaked the VAE architecture, input image size and the number of latent dimensions in order to improve the interpret-ability of the model. Below figures show the output of various models.

Figure 13. Grad-CAM generated heatmaps of individual latent nodes of latent space of size 5 and 64x64 input images.

When a small latent space dimension is chosen, each dimension is overloaded, and it has to embed large part of the image and thus interpretability is reduced, which can be

seen in figure 13. To counter this issue, we increased the size of the latent dimension to 15 and retrained the new model. Increasing the latent space dimensions to 15 increased the distribution of the features of the images in latent space, but the heatmaps remained random as shown in figure14.

Changing the latent space dimensions to 100 and taking 256x256 images and applying Grad-CAM to the model produces heatmaps which show that individual dimensions focus on specific facial features of the images, figure 15 highlight the hair part of the image for 3 different input images. However, when explored other latent dimensions, it is discovered that the latent space embedding do not always correlate across different images as shown in the figure 16.
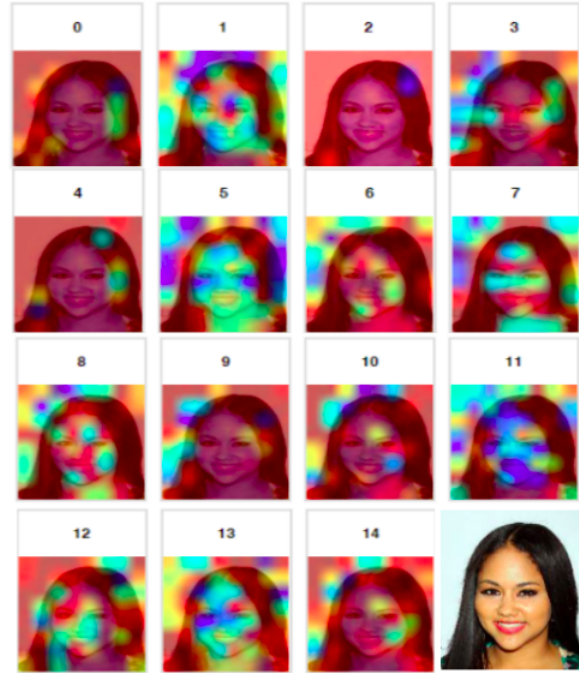

Figure 14. Grad-CAM generated heatmaps of individual latent nodes of a latent space of size 15 and 64x64 input images.
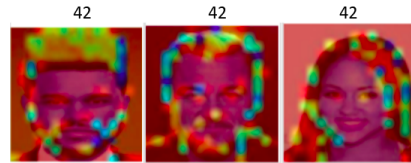

Figure 15. Grad-CAM generated heatmaps for latent node 42nd of the latent space of size 100, for three different input images.

5.4. Conclusion and Future Scope

In our work we attempted to analyse the activation map of a generative model, by applying Grad-CAM and tweaking the latent space as discussed in sections 4 of this paper, we assert three major findings.
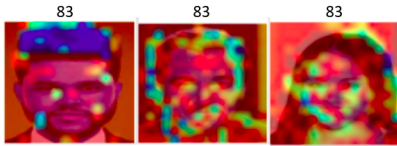
Figure 16. Grad-CAM generated heatmaps for latent node 83rd of the latent space of size 100, for three different input images. The first heatmap is focusing on the hair, but the remaining 2 heatmaps are covering the face structure and hair, this is 100 dimensional latent space and 256x256 input image model.

First, The activation maps of a VAE encapsulates information about different features of the image data. This can be clearly seen in section 5.1, which shows the averaged heatmaps for all latent space nodes. and also with help of the individual node heatmaps.

Second, the individual latent space nodes store multiple feature information required to recreate images, as tweaking the value of a single node led to a change in different features of the generated image.

Third, Semantic Segmentation works well with VAE models and the output is comparable with that of the U-net (Vinogradova et al., 2020). Smaller latent space inhibits the interpretability of the model, whereas larger latent space distributes the features of the image in the latent space and only few dimensions are correlated.

As for the future scope of this work, a systematic approach to choose the number of latent space dimensions and a metric to evaluate set of most significant latent nodes should be investigated.

Software and Data

CelebAMask-HQ is a large-scale face image dataset with 30,000 high-resolution face images along with 512x512 size segmentation mask for all facial components such as skin, ear, eyes, eyebrows, ears, mouth, lip, hair, hat, eyeglass, earring, necklace, neck, and cloth. We have used this dataset to train a VAE to generate facial images and face segmentation masks. Each image has 40 binary attribute annotations, which was used for training a Conditional VAE model.

References

Aggarwal, A., Mittal, M., and Battineni, G. Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, pp. 100004, 2021.

Albawi, S., Mohammed, T. A., and Al-Zawi, S. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6. Ieee, 2017.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

Chakraborty, S., Tomsett, R., Raghavendra, R., Harborne, D., Alzantot, M., Cerutti, F., Srivastava, M., Preece, A., Julier, S., Rao, R. M., et al. Interpretability of deep learning models: A survey of results. In *2017 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation (smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI)*, pp. 1–6. IEEE, 2017.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Fodor, I. K. A survey of dimension reduction techniques. Technical report, Lawrence Livermore National Lab., CA (US), 2002.

Grzywaczewski, A. Training ai for self-driving vehicles: the challenge of scale. *Available from Internet: https://devblogs. nvidia. com/training-self-driving-vehicles-challenge-scale*, 2017.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.

Holzinger, A., Biemann, C., Pattichis, C. S., and Kell, D. B. What do we need to build explainable ai systems for the medical domain? *arXiv preprint arXiv:1712.09923*, 2017.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Kingma, D. P. and Welling, M. An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*, 2019.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. *Advances in neural information processing systems*, 28:2575–2583, 2015.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

Liddy, E. D. Natural language processing. 2001.

Liu, W., Li, R., Zheng, M., Karanam, S., Wu, Z., Bhanu, B., Radke, R. J., and Camps, O. Towards visually explaining variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8642–8651, 2020.

Parker, J. R. *Algorithms for image processing and computer vision*. John Wiley & Sons, 2010.

Rai, A. Explainable ai: From black box to glass box. *Journal of the Academy of Marketing Science*, 48(1):137–141, 2020.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Szeliski, R. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

Vinogradova, K., Dibrov, A., and Myers, G. Towards interpretable semantic segmentation via gradient-weighted class activation mapping. *arXiv preprint arXiv:2002.11434*, 2020.

Zaremba, W., Sutskever, I., and Vinyals, O. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.