Objective Evaluation on the Performance of General Adversarial Neural Networks

Alexander J. O'Donnell, Patrick J. M. Savoie

University of New Brunswick, Department of Electrical and Computer Engineering, Dec 4th, 2017

Abstract*— Generative Adversarial Networks, or GANs have been used to create a generative artificial intelligence system such that the reproductions of the model accurately represent the distribution of the original samples used during a training session. A notorious problem in Deep Convolutional Neural Networks, (DCGANs), is their instability during training due to gradients converging at local minima. The problem here is that the evaluation for the generative network and the discriminative network both consist of neural networks. This is effectively black boxing the system. At this point it becomes difficult to identify problems that occur during training. We propose a means to objectively evaluate the performance of a generative network using more intuitive pattern recognition techniques such as Linear Discriminant Analysis, PCA thresholding K Nearest Neighbors and Support Vector Machines. Dimensionality of the problem was also significantly reduced in two different ways though Principal Components Analysis, and Fisher Linear Discriminant Analysis. When tested on a DCGAN model that generates human faces, the result is a set classifiers that can be used in order to weed out generated images that poorly represent a natural face. With this quality metric established, it is now possible to implement rejection in outliers of the DCGAN's model distribution.

Key Terms*— Fisher Linear Discriminant Analysis, Generative Adversarial Networks, Pattern Recognition, Principal Components Analysis.

I. Introduction and Background

A. Generative Models and Gap

The promise that comes alongside artificial intelligence is the development of rich models that apply to problems in real life. The field has blown up in recent years as computation power catches up to the expensive algorithms required to tackle these problems. In addition to this, the internet has allowed for the precipitation of unfathomable pools of data. An enormous issue with this information prosperity, is the difficulty of sifting through all the nonsense and making it useful in applications for artificial intelligence. In recent years, generative models have been used to capture vast amounts of data within a statistical distribution. With respect to generative models, most of the existing work comes in the form of Neural Networks. Many different architectures are being used such as Auto Encoders, Recurrent Neural Nets, Convolutional Neural Nets and Generative Adversarial Nets. This issue arises where a large portion of understanding is lost through the use of this style of machine learning. Because of this, generative networks are practically impossible to debug and prone to gradients converging at local minima. The problem with Generative Adversarial Networks is that the generator and discriminator both rely on artificial neural networks. A gap in current research can be identified in the use of more intuitive, statistically heavy, pattern recognition systems as an objective evaluation for the performance of these networks. Currently, evaluation of these networks has been subjectively observing the results. We propose, that through the use of more intuitive pattern recognition, there is a way to perform this evaluation. Techniques such as Principle Components Analysis, Linear Discriminant Analysis and Support Vector Machines will be used in this paper to do exactly that.

B. Current Strides in Generative Adversarial Networks

There has been a lot of effort placed into the techniques in the production of models that can be used to accurately represent the data encountered in artificial intelligence applications. The promise of accurately capturing the statistical distribution of a given dataset, is increased performance of the model in solving problems. Currently, generative models are being used to infer the high dimensional structure of data [1]. Generative techniques are also extremely useful when it comes to establishing labels for vast amounts of data using weak supervision [1, 2]. In contrast, discriminative models have shown incredible success in connecting high dimensional data to a member of a set of class labels. In recent years, Generative Adversarial Networks, or GANs [2] have been using a generator and a discriminator in order to create completely original data that is representative of the training set. The model is often explained as an artist, and an art critic that work together in order to provide completely artificial samples that are similar to real instances of a dataset.

In a Deep Convolutional Generative Adversarial Network (DCGAN), the generator and discriminator consist of a convolutional and a deconvolutional neural network [3]. The generator starts out with a random noise vector and deconvolves it into an approximation of the training data that falls into the space of the current model. This generated sample is then fed through the discriminator and sent through a series of convolutional filters that map the image to a single value. During training, this value is used to evaluate the performance of the DCGAN. A popular experiment with DCGANs seen in recent years the generation of completely original images through unsupervised methods. Trained generator models have had striking results when it comes to creating instances that fall into the distribution of the original data.

II. Methods

A. Training the DCGAN

For the purposes of the experiment, a DCGAN was trained in order to produce images that reflect samples from a large set of faces. The instances of faces were retrieved from the CelebA dataset, a group of 202,599 images available for non-commercial research purposes from the Multimedia Laboratory (MMLAB) at the Chinese University of Hong Kong [4]. It also should be stated that the samples were simply

obtained from the internet by MMLAB therefore they are not actually the property of the University. The Matlab source for the DCGAN used during the procedure was heavily adapted from a GitHub user that goes by "sunghbae" [5]. Their implementation uses the MatConvNet convolutional neural network library in order to set up the model. The software is free to be adapted by anyone at their own risk under an MIT license. The random seed vector used to generate images was also set up using the "shuffle" random seed in Matlab. This type of random seeding uses the date and time. This means that once a model is trained, artificial faces in can be generated indefinitely.

B. Preprocessing

With the DCGAN architecture set up, and the CelebA dataset gathered, some measures were made in order to ensure that the data would be effective in building a pattern recognition system. First, the celebrity images were cropped and aligned such that the distribution contained as much face as possible. The purpose of this is to minimize the percentage of the image composed of background. Without this step, the generator may have a hard time recreating the entire image since there would not be enough consistency in the background of the samples to construct a good model of what that portion of the image should look like. After the faces were center aligned and cropped, they were all compressed into 64 by 64 pixel images. The consistency in resolution is key as the pixels will be used as features for training the DCGAN. In addition to this the pixels of the image will become the initial set of features used before dimensionality reduction and feature extraction. It also should be stated that colour images were used for the DCGAN while the images were converted to grayscale during dimensionality reduction. The reason the grayscale images were used for PCA and Fisher LDA is because there wasn't any literature supporting principle components for coloured image. Also, images in colour contain similar information in all 3 of the RGB channels. For this reason, it was concluded using coloured images for dimensionality reduction would be redundant. In a sense, converting to grayscale is a form of dimensionality reduction itself.

In addition to the CelebA faces gathered in order to train the GAN, non-face images were retrieved from the CIFAR-10 dataset [6]. Any outlying face that was a part of the dataset was manually deleted in order to assure that no faces made it into the set. With all positive instances of a face removed, the CIFAR-10 set consisted of around 48000 instances of seemingly random images such as large mammals, household objects, flowers, apes and more. The importance of these images will be realized when the classifiers are constructed. The non-face class will be the class of "non-faces" during the binary classification problem.

The last dataset that should be mentioned, is the artificial faces generated from the DCGAN. These images will be evaluated by the binary classification models to determine if an image is a face or not. Since the images were generated as 64 by 64 pixel images, the first preprocessing step required was to convert the set into grayscale. Different approaches were used to determine the performance of the GAN, PCA thresholding, and various classifiers, both of which required additional, and slightly different pre-processing techniques.

C. Measuring Performance of the DCGAN

1) Thresholding with Principal Components Analysis

One of the methods used to determine the accuracy of the GAN was a PCA threshold [7, 8]. The basis for this approach is that the principle components that represent most of the variance for images of faces do not represent most of the variance of non-face images, and as such, do not reconstruct these images well [7]. The steps for this method are as follows: A dataset of faces is found, in this case the CelebA Dataset. The mean image is computed and subtracted from each image. Next, the covariance of the mean aligned images is calculated. An orthogonal matrix of the eigenvectors, and a diagonal matrix of eigenvalues of the covariance matrix is subsequently found, where the eigenvalues are sorted in descending order, and the matrix of eigenvectors are sorted to their corresponding eigenvalue. Each column in the matrix of eigenvectors corresponds to a new basis to represent the image, termed "Eigen Image". To reduce the dimensionality, the eigenvectors that correspond to the majority of the variance are kept, while the others are discarded. In this work, the variance was set to 98 percent. New images are subsequently projected on the face space through Equation 1). The original image are subsequently reconstructed using Equation 2), but there will be error in the reconstructed image due to the fact that several basis images were discarded. Equation 3) is used to calculate the reconstruction error. The error of non-face images should be higher than the images of faces, therefore a threshold can be set on the maximum tolerable reconstruction error for the input image to be considered a face [7]. This threshold is often determined heuristically, however in the case of this work, it was determined as the maximum reconstruction error of a subset of the images in the dataset. This was done for the results to be reproducible with different datasets. Due to memory limitations, PCA was performed on 50000 images of the CelebA dataset, and the threshold was set as the maximum reconstruction error of these 50000 images.

$$\Omega = \mathrm{U}^T(\Psi - \mathrm{u}) \qquad (1)$$

Where Ω is the new representation of the image, Ψ is the input image, u is the mean image, and U is the matrix of the principle components containing 98 percent of the variance of the data.

$$\Psi^{rec} = \mathrm{U}\Omega + \mathrm{u} = (\textstyle\sum_i U_i * \Omega_i) + u \qquad (2)$$

Where Ψ^{rec} is the reconstructed image, the reconstructed error is subsequently calculated as:

$$e = \|\Psi - \Psi^{rec}\|_2 \qquad (3)$$

Another method to determine if the generated images were faces or not was through classifiers and labelled data. Several common classifiers for face recognition include Linear Discriminant Analysis classifiers (LDA), Support Vector Machines (SVM), and K Nearest Neighbors (KNN) [9, 10]. In this work, these classifiers are used to classify the generated faces as faces, or non-faces. To this end, the

classifiers are trained on two classes of images: faces and non-faces. The face images are provided by the CelebA dataset, and the images of non-faces are used from the CIFAR-10 dataset. After pre-processing, approximately 48000 images of non-faces remained from the CIFAR-10 dataset, and these images were used as the non-face images. 50000 images from the CelebA dataset was used as the images of faces. The dimensionality of the input images is 4096, which is extremely high, therefore projection techniques were used to reduce the dimensionality of the data to reduce the effects of the curse of dimensionality.

D. Dimensionality Reduction

Common techniques for dimensionality reduction include PCA and Fisher Linear Discriminant Analysis. PCA reduces the dimensionality by projecting the data on the vectors which capture the most variance of the data, and discarding the rest. In this work, this number was set to 98% of the variance. It should be noted that these principle components are different from those found through PCA thresholding, as the ones found through the thresholding method represent the principle components solely of faces. In this new method, the principle components of a combination of face and non-face images is found.

1) Through Classifiers and Fisher LDA

Fisher Linear Discriminant Analysis, or Fisher LDA is a supervised technique for dimensionality reduction that relies on a projection from a high dimensional space to a lower dimension such that interclass variance is minimized and infraclass variance is maximized. In the case of a binary classification problem, (is it a face or is it not a face), Fisher LDA will reduce the 4096 pixels in the images down to a single dimension that provides the optimal separation between classes. The benefit to this is speed in training a classifier. One small downside is that any new sample will need to be transformed into the same dimension before it can be evaluated.

The Classifiers are trained with the data's new feature representation with 10 fold cross validation. In the case of PCA, where the number of features is greater than 1, the classifiers are trained with various amounts of principal components, and the maximum number of features is the amount of principal components representing 98% of the variance. The parameters of KNN and SVM significantly affect the accuracy of the classifier, therefore they need to be tuned to obtain optimal accuracy. In the case of KNN, the value of k was varied from 1 to 30, and the distance metric was iterated between Euclidean, Chebyshev, and Manhattan. In the case of SVM, several kernels were used, including a linear, Gaussian, polynomial, RBF kernel. After training the LDA classifier on both the PCA and Fisher LDA projected data, it was found that Fisher LDA had higher accuracy, and a computation time of several orders of magnitude less than PCA (Less than 1 second to train LDA features, versus approximately 15 minutes for PCA features), therefore PCA features were not tested further with KNN and SVM, as these classifiers would have required several days to train and tune their parameters, when LDA features already produced sufficient accuracy for the purposes of this work, with a significantly lower run time. Once the classifiers were trained, the generated images were given to the trained model to determine how many were considered to be faces, and non-faces.

III. Results and Discussion

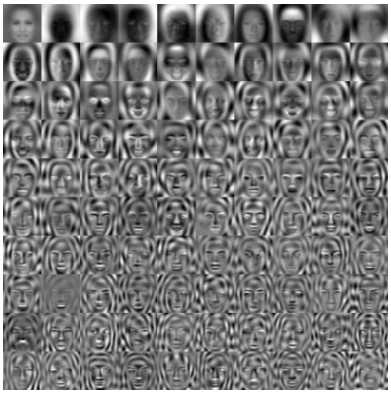
A. Performance of the DCGAN

Once the DCGAN model was trained over several days, the capacity to generate completely original samples was established. For the purpose of this experiment, 5000 images were selected for evaluation. Subjectively, the generated images varied significantly. In terms of the empirical pattern recognition system that all humans come equipped with, a lot of these images would fail the test. It also should be noted at this point that he generative network had a hard time generating individuals wearing hats or sunglasses. This makes sense given the fact that there weren't too many samples in CelebA of this occurring. Without the proper supervision, the DCGAN struggles to map the areas in the pixel space that correspond to these characteristics. Other instances of the artificial faces appear lifelike and indistinguishable from a real person, even to the human eye. With the final artificial face dataset prepared and visually analyzed, it was time to see how the machine performs in objectively evaluating the appearance of these fake samples. A random subset of the images can be seen in Figure 1.

Figure 1. Subset of Artificial Human Faces

B. Thresholding Images with Principal Components Analysis

Figure 2. First 99 Eigenfaces



After performing PCA on the celebrity images, and keeping the components corresponding to 98 percent of the variance of the images, 522 Eigenfaces remained. The first 99 Eigenfaces are illustrated in Figure 2, with the first image in the montage representing the mean face. PCA thresholding resulted in 63 out of the 5120 generated images not being recognized as faces, corresponding to an accuracy of 99.08 for the generated images. The images that failed the PCA thresholding test, by having a higher reconstruction error than any of the 50000 images used from the CelebA dataset, are shown in Figure 3. These images are plotted in descending order of reconstruction error (from left to right, and the last image being on the bottom right). Most of these generated images appear to be high noise, and include either hats or glasses, or do not resemble a human face whatsoever. This is expected, because the majority of the images from the CelebA do not have hats or glasses, therefore the principle components do not reliably reconstruct such images. It should be noted that many of the images that failed PCA thresholding look identical at a first glance, however taking the difference between the images reveals that these images different, they just look similar. The images with the minimum reconstruction error are plotted in Figure 4. These images appear to be significantly lower noise than those which failed the test. Some of these images do not appear to resemble faces, or at least particularly good looking faces, which illustrates how PCA generates features which are not immediately obvious to humans. It should also be noted that due to the hard thresholding nature of this method, several images that were close to the threshold were accepted as faces. These images are shown in Figure 5, and are pretty similar to the rejected images. There was no literature supporting PCA thresholding with soft thresholding, or with a percent confidence, however if such a method had to be suggested no additional method was incorporated to address the issue of hard thresholding of PCA.

Figure 3. Faces that Fail PCA Threshold

Figure 4. Faces that Perform Best on PCA Threshold

Figure 5. Faces that Almost Fail PCA Threshold

C. Classification Using PCA

After performing principal component analysis on the data of faces and non-faces, 565 principle components remained as the lower dimensional representation. The classifier was trained with varying amounts of features, up to 565, and a achieved maximum mean accuracy of 96.7% from 10 fold cross validation with 565 principle components. The inflection point of the accuracy-feature curve is at around 60 features, therefore further increases lead to minimal increases in accuracy. These results are plotted in Figure 6. The optimal classifier classified 279 of the GAN generated images as non-faces, meaning 94.55% of the GAN images were considered as faces by this classifier. The first 64 rejected faces are plotted in Figure 7. This classifier appears to perform better than the PCA thresholding method, as the classifier is not rejecting images based on if it includes a hat, and glasses, most of these images simply do not look like faces based on simple observation.

KNN and SVM classifiers were not attempted with PCA features, since Fisher LDA features yielded superior accuracy for the LDA classifier, and had a run time of several orders of magnitude less than PCA features. KNN and SVM would have taken almost a week of continuous run time to train, and to optimize their parameters; however, with fisher features this training was able to take place in less than a day, and the accuracy was found to be high for both KNN and SVM classifiers using Fisher LDA features.

Figure 6. Distribution of Faces and Non Faces Plotted in 1D

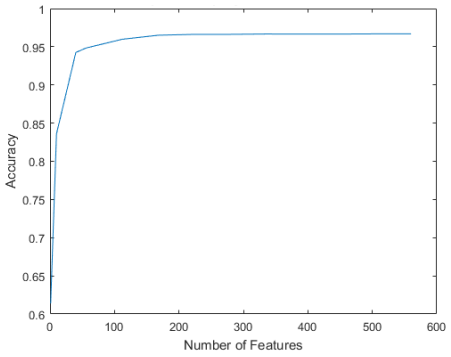
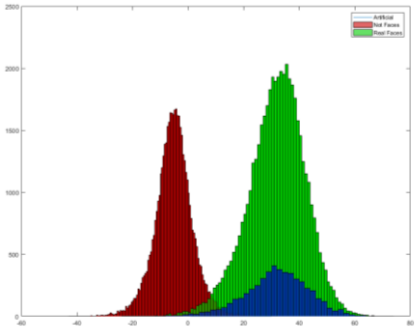


Figure 7. 64 Faces Regected by Classification through PCA

D. Classification Using Fisher LDA

As mentioned in the methods section, with a binary classification problem, Fisher LDA can reduce the dimensionality of the problem to a single feature. For the 1D model to be established, Fisher LDA was trained using 96000 instances of grayscale images. The reason the set was constrained to this many samples is because it was desirable that there was an equal parts faces and not faces, and there were only 48000 non face images available. A histogram was generated in order to show the distribution of faces versus non faces in the new singular feature space. The left most Gaussian, shown in red, represents the mapping of the non-face images to the space. On the right, the green cluster describes the distribution of the face images. The smallest Gaussian, shown in blue, consists of 5000 artificial faces mapped to the 1D space using the transform matrix retrieved from Fisher LDA. From mapping artificial faces to the new feature space, it is clear that these images are representative of the real faces that undergo the same transformation. Subjectively, based on the overlap of distributions of the real and generated faces, this appears like the DCGAN has successfully generated counterfeit faces that register as real ones. With this in mind, objective classification metrics were established in order to let the machine evaluate the generated Faces.

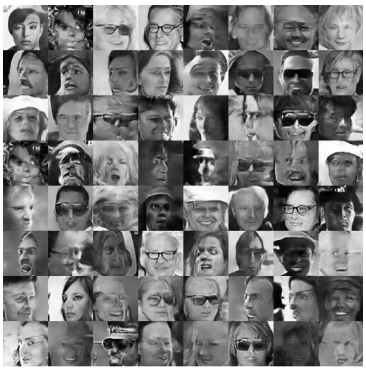
Figure 8. Distribution of Faces and Non Faces Plotted in 1D



For this classification problem, LDA, KNN and SVMs were used in order to examine the quality of the generator. Training and test partitions were established using k-fold cross validation where k = 10. The mean classification accuracy across all partitions was then used as a metric in order to evaluate the generated image. The hypothesis at this point becomes, if the image does not reflect a real human face, it should fail to classify as one. A poor model representation of the original samples will fail to fall in the same class

With LDA, the model constructed had an accuracy of 97.3% when it comes to classifying faces and non faces. With the artificial faces are transformed into a representation in the 1D feature space and classified using the model, 287 samples fail the test. Once again, the performance of this model was observed subjectively analyzed by displaying the first 64 images that fail to be classified as people. Most of these images don't look anything like a human face. A lot of the faces are either extremely noisy, or demonstrate outlying characteristics that the DCGAN has difficulty modelling such as glasses or hats.

Figure 9. 64 Imaged that failed LDA

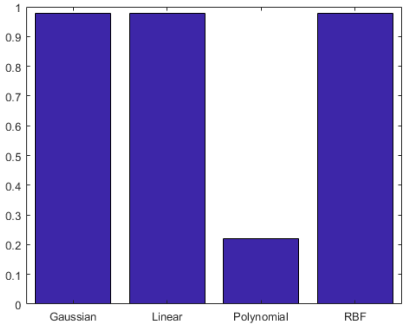


When the KNN classifier was trained, it was determined using different K values and distance metrics. As far as distance metrics, Euclidean, Chebyshev and Manhattan distances were all found to yield the same accuracy. This makes sense given the simplicity and distribution of different classes in the feature space. The faces and non faces represent 2 Gaussian densities with very little overlap therefore distances will appear the same independent of the metric used. The optimal value of K was determined to be 16 for every metric. It appears that an increased K past this point will result in diminishing returns, however if K is increased past a certain point, the classifier will reflect the prior probability of our dataset which is 50% faces and 50% non faces. Once the optimal values were determined, KNN was used to generate our final model. With K = 16, the average accuracy across every partition was 97.85%. This was achieved using a Euclidean distance metric, however, both Chebyshev and Manhattan distances did have similar performances with accuracies in the upper echelon of 97%.The KNN classifier managed to discard 197 generated samples in total. These instances held similar characteristics to the set of images rejected by LDA in the sense that most of them had glasses or high noise content. Of the first 64 rejected samples, there is some overlap with the 2 rejections. One notable difference in the rejections of the 2 classifiers, is the lady in the top left hand corner that looks like a good reproduction.

Figure 10. 64 Imaged that failed KNN

For the support vector machine different kernels were used to emulate the data in a higher dimension. Out of a set of linear, Gaussian, RBF and polynomial kernels, all of them provided similar results except for the polynomial kernel with a mean accuracy of only 22.11%. For the other Kernels, SVM yielded similar mean accuracies around 97% with the linear kernel performing the best at 97.83%. A bar plot was created in order to demonstrate the accuracy of all the kernels.

Figure 11. Accuracy of Different SVM Kernels



In total, 186 images were rejected using the optimal SVM kernel. Again we can trust our developed pattern recognition system to validate that the samples discarded are not good representations of the original dataset. Most of the images are either extremely noisy or attempted reproductions of glasses or hats. In comparison to the first 64 rejected images of the other classifiers, members in the set of images rejected by the SVM are an intersection of the other two classification models. A table was constructed in order to summarize the results.

Figure 12. 64 Imaged that failed SVM

TABLE I. CLASSIFICATION WITH OPTIMAL PARAMETERS

Method	**Performance**	
	Mean Accuracy	***Standard Deviation***
LDA with PCA components	0.9670	0.0023
LDA with Fisher features	0.9730	0.0026
KNN with Fisher features	0.9785	0.0012
SVM with Fisher features	0.9783	0.0015

IV. Conclusion

In the end, the evaluation of generated faces objectively through the construction of intuitive pattern recognition systems was a success. Various methods were used to evaluate the generated faces, including a PCA threshold, and various classifiers such as LDA, KNN and SVM, statistical approaches were used in order to establish classifiers on the images as they lie in the single dimension created though Fisher LDA. When artificial samples were transformed and mapped onto this dimension, a lot of images fell into the same distribution as real faces. Once classifiers were trained using this information. The greatest accuracy for classifying faces and non faces was a mean of 97.85% across all partitions achieved through KNN with Euclidean distance, and K = 16 on the Fisher feature space. The SVM and LDA classifiers performed exceptionally as well with mean accuracies of 97.83% and 97.3% respectively. It also should be stated that LDA rejected the most when it comes to artificial images. At a first glance, this may appear desirable in the sense that fewer generated faces make it though, however, some of the faces rejected are good representations of the training data. Revisiting the example of the first girl that was rejected through LDA, she holds the characteristics of other images that get rejected such as accessories or high noise content. Overall, this work can be used to filter undesirable images from a GAN, and display to the user the images that are most desirable to see (Those which look the most like humans). One limitation of this approach, is that many of the rejected images are rejected solely based on the fact that they contain hats, or sun glasses. These images are occasionally removed unnecessarily, even though most of the image looks like faces, which could be undesirable for an end user.

As far as future work with GANs, there is a lot that is left to be done. First off, it would be nice to gather a dataset with more samples of hats and glasses in order to see if the DCGAN could learn to replicate these characteristics. In addition to this, it would be interesting to see how a hybrid model consisting of a deconvolutional neural network as a generator and one of the classifiers used as a discriminator would fare at generating artificial images. For example, the generator could be rewarded for reproducing a sample that can be successfully reconstructed using the principal components of real faces. It would also be interesting to see how a poorly trained GAN performs against the classifiers built during the experiment. The DCGAN that was implemented was trained for several days so good performance from the system was expected. The field of Machine Intelligence screams the promise of rich models that can be used in order to solve limitless problems and it's incredibly exciting.

V. References

[1] P. Varma, B. He, Payal Bajaj, Imon Banerjee, Nishith Khandwala, Daniel L. Rubin, and Christopher Ré, "Inferring Generative Model Structure with Static Analysis," Inferring Generative Model Structure with Static Analysis, Sep. 2017.

[2] Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron C., and Bengio, Yoshua. Generative adversarial nets. NIPS, 2014.

[3] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." arXiv preprint arXiv: 1511.06434 (2015). Available: https://arxiv.org/abs/1511.06434

[4] Deep Learning Face Attributes in the Wild, Proceedings of International Conference on Computer Vision (ICCV), Ziwei Liu, Ping Luo, Xiaogang Wang and Xiaoou Tang, University of Hong Kong, December 2015.

[5] Sunghbae, Deep Convolutional Generative Adversarial Network (DCGAN) implementation on MatConvNet, (2017), GitHub repository, Available: https://github.com/sunghbae/dcgan-matconvnet

[6] Alex Krizhevsky, CIFAR-10 Dataset, Learning Multiple Layers of Features from Tiny Images, 2009. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[7] F. Jalled, "Face Recognition Machine Vision System Using Eigenfaces," Moscow Institute of Physics & Technology, May 2017.

[8] M. Turk and A. Pentland, "Face recognition using eigenfaces," Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[9] P. Parveen and B. Thuraisingham, "Face Recognition using Multiple Classifiers," Jun-2006.

[10] A. Bouzalmat, J. Kharroubi, and A. Zarghili, "Face Recognition Using SVM Based on LDA," IJCSI International Journal of Computer Science Issues, vol. 10, no. 4, Jul. 2013.

[11] S. Alzahrani, "Matlab-Source Code-An-Implementation-of-the-Fisher-Discriminant-Analysis 1.1," Sep. 2016.

12] Telgaonkar H Archana, Deshmukh Sachin, Dimensionality Reduction and Classification through PCA and LDA, Available: http://research.ijcaonline.org/volume122/number17/pxc3905104.pdf, July 2015,