COMPUTERS AND INFORMATICS
2757-8259 2021, Volume 1 Issue 1
Research Article

Fake human face recognition with classical-quantum hybrid transfer learning

Furkan Ciylan
OSTIM Technical University, Ostim, 100. Yıl Blv 55/F, 06374, Ankara, Turkey
Bünyamin Ciylan
Gazi University, 06560, Ankara, Turkey

Submitted: 19.02.2020
Accepted: 20.02.2020
Published: 28.02.2021

Abstract:

Many security applications like face recognition and iris recognition developed to ensure the safety of the critical or personal data. Reliability of these applications are highly depending over the reliability of machine vision algorithms. Along with the development of the generative models standard machine vision dependent data security measurement systems became vulnerable. Currently, generating fake data to bypass machine vision dependent security systems is possible with a personal computer. In order to ensure the reliability of security measurements depending on the machine vision techniques it is critical to recognize fake images. In this research, possible use case of a quantum computer to ensure the reliability of machine vision dependent security systems is investigated. A hybrid quantum-classical hybrid model with the transfer learning approach is proposed to recognize whether if a face is fake or not. Effects of the quantum model's depth over the accuracy is explored. ResNet-18 architecture is used as the classical part and a custom quantum neural network architecture built with the dressed quantum circuits is used as quantum part. This research is aimed to extend the use cases of quantum neural networks to security applications. Accuracies of quantum neural networks with different depths are reported. A simulated quantum computer is used to train the models. Along with the proposed approach it is concluded that it is possible to apply classical-quantum neural networks to improve the reliability of machine vision dependent security systems after the quantum co-processors become available in daily life.

***Keywords**: Deep learning, Fake image, Image classification, Transfer learning, Quantum, Quantum machine learning*

Cite this paper as: Ciylan F., Ciylan B. Fake human face recognition with classical-quantum hybrid transfer learning, *Computers and Informatics*, 2021, *1*(1), 46-55.

1. INTRODUCTION

Recently, Deep Learning algorithms are integrated with the machine vision technologies. Artificial Neural Networks made applications like object detection, image classification, object tracking and synthetic image generation more efficient and accurate. Many state of the art results are achieved with the recent development in Deep Learning in image processing [1]. Also, deep learning algorithms used to create synthetic human faces in a normal looking image. This technology is a potential threat to the security systems depending on the face recognition. Thus, face recognition systems should also understand whether if an image is fake or real.

Convolutional Neural Networks (CNN) are mainly focused on the machine vision applications. Their working principle is inspired from the pattern of interconnection between neurons similar to the organization of the animal visual cortex which proposed by Hubel and Wiesel [2]. It is possible to describe the general architecture of CNNs as one input layer, one output layer and the hidden layers. A hidden layer in a CNN could be a convolutional layer, pooling layer, batch normalization layer or fully connected layer. The working principle of a convolutional layer is like a reaction of a neuron in the visual cortex to a specific stimulus. These layers are consisting of filters to create a feature map in order to detect features of an input. Pooling layers are consisting of kernels in order to reduce the size of an image by using the output features of convolutional layer. Batch normalization layers are used to prevent over-fitting. They add specific amount of randomness to the architecture. Fully connected layers are the layers where all neurons in a layer are connected to all neurons in the next layer. Generally, using a fully connected layer instead of a convolutional layer is possible. But it would require many connections between neurons. Instead by using a convolutional layer, creating a deeper network with less parameters with decreasing amount of free parameters in a network is possible.

Quantum neural networks are using principles of the quantum mechanics in order to improve neural network architecture. For improving the architecture, quantum parallelism and effects of quantum entanglement, quantum interference features of quantum computing are used as resources. Mostly, a quantum neural network is used along with a classical neural network to improve the performance of the general network. Training and testing these hybrid architectures on IBM Quantum Experience by using quantum computers with IBM Qiskit is possible. Also, simulating quantum circuits to train and validate quantum-classical neural network architectures on classical computers could be very useful for most cases. There are many limitations about quantum neural networks, related to the continuing development of quantum computers. There are only a limited number of available quantum computers in world. Thus, many works are for research purposes only.

In this work, several quantum-classical hybrid neural network architectures are developed and tested to recognize fake human face images. This work is mainly focused on understanding the effect of a quantum computer on fake image recognition. It is important to understand relationship between accuracy and architecture of a quantum-classical hybrid neural network’s quantum neural network part in fake human face recognition to develop a methodology for future quantum-security applications.

2. RELATED WORK

2.1. Quantum Neural Networks

Along with the current developments in Quantum Computing, now it is possible for a Quantum computer to solve problems which is almost impossible to solve for many classical computers [3][4][5]. A realistic near-term quantum device [6] has a potential to find a solution for many optimization and sampling problems by handling memory and time consuming operations for a classical computer.

Lewestein, has introduced the first quantum based perceptron [7]. In Lewestein's research a unitary operator used to map inputs and outputs of a classical perceptron to use advantages of quantum computing. Then Chrisley introduced a practical quantum learning model [8][9]. But this approach does not test with a simulation instead Chrisley used double slit equipment for explaining a feed-forward quantum artificial neural network. Chrisley's approach has been extended by Menneer and Narayanan for designing a single pattern quantum neural network [10][11]. In 2018, Pierre-Luc Dallaire-Demers and Nathan Killoran has proposed the Quantum Generative Adversarial Networks [12]. They used quantum circuits to create a generator and discriminator. They successfully generated fake data by using a Quantum Generative Adversarial Network. In 2019, Nathan Killoran et al. proposed a continuous variable quantum neural network [13].

Nowadays, it became possible to use quantum neural networks as a part of a classical neural network like a co-processor. By using a Quantum-Classical Hybrid network some can use the advantages of quantum supremacy only in some particular operations in a classical neural network. A hybrid quantum-classical neural network has been developed to calculate ground state energies of molecules as an example. [14]. A model to use transfer learning approach in quantum-classical hybrid neural networks has been proposed by Andrea Mari et al. [15].

3. MATERIALS AND METHODS

3.1. Proposed Model

3.1.1. Classical neural network

A pre-trained residual network (ResNet-18) model is used as classical part of a Quantum-Classical Hybrid Neural Network in this research. ResNet-18 is a deep convolutional neural network with pooling, convolutional, fully connected and softmax layers.

In order to define an image as a tensor eq. (1) is used.

$$\dim(image) = (n_H, n_W, n_C) \tag{1}$$

Where n_H is the size of height, n_W is the size of width and n_C is the number of channels. A filter should have same number of channels with the image. Thus, in order to define a filter, it is possible to use eq. (2).

$$\dim(filter) = (f, f, n_C) \tag{2}$$

Convolutional product between an image and a filter is a 2D matrix. Every value in this matrix is the result of sum of elementwise multiplication of filter and representation of image in each channel. It is possible to get convolutional product by using eq. (3).

$$\operatorname{conv}(I, K)_{x,y} = \sum_{i=1}^{n_H} \sum_{j=1}^{n_W} \sum\nolimits_{k=1}^{n_C} K_{i,j,k} I_{x+i-1,y+j-1,k} \tag{3}$$

In order to find the dimensions of a convolutional product between an image and a filter (4) is used by considering the eqs. (1), (2) and (3).

$$\begin{aligned}\dim\left(\operatorname{conv}(I,K)\right) &= \left(\left\lfloor \frac{n_H+2p-f}{s}+1\right\rfloor, \left\lfloor \frac{n_W+2p-f}{s}+1\right\rfloor\right); s>0 \\ &= (n_H+2p-f,\ n_W+2p-f); s=0\end{aligned} \tag{4}$$

Where p is padding and s is stride.

It is possible to use pooling operation to down sample the features of an image. Pooling process does not affect the number of channels n_C but instead it affects only n_H and n_W. It is possible to define pooling operation as (5).

$$\begin{aligned}\dim\left(\operatorname{pooling}(image)\right) &= \left(\left\lfloor \frac{n_H+2p-f}{s}+1\right\rfloor, \left\lfloor \frac{n_W+2p-f}{s}+1\right\rfloor, n_C\right); s>0 \\ &= (n_H+2p-f,\ n_W+2p-f, n_C); s=0\end{aligned} \tag{5}$$

Two widely used types of pooling are: Max pooling and average pooling. When using the max pooling only the maximum value in a filter will return. In average pooling average value in a filter will return.

Many convolutional products are applied along with the many filters in a convolutional layer. A mathematical representation of a convolutional layer is shown in eq. (6) where $\forall n \in [1,2,3, \ldots n_c^{[l\}}]$

$$\operatorname{conv}(a^{[l-1]}, K^n)_{x,y} = \psi^{[l\}}\left(\sum_{i=1}^{n_H^{[l-1\}}} \sum_{j=1}^{n_W^{[l-1\}}} \sum\nolimits_{k=1}^{n_C^{[l-1\}}} K_{i,j,k}^{n} a_{x+i-1,y+j-1,k}^{[l-1]} + b_n^{[l]}\right) \tag{6}$$

Where l is the layer as an integer value, $a^{[l-1]}$ is the input with the size of $(n_H^{[l-1]}, n_W^{[l-1]}, n_C^{[l-1]})$, padding is the $p^{[l]}$, stride is $s^{[l]}$, $\psi^{[l\}}$ is the activation function, $b_n^{[l]}$ is the bias at nth convolution, $n_C^{[l-1]}$ is the number of filters where K^n has a dimension of $(f^{[l]}, f^{[l]}, n_C^{[l]})$. It is possible to represent the dimensions of a convolutional layer as eq. (7).

$$\dim\left(\operatorname{conv}(a^{[l-1]}, K^n)\right) = (n_H^{[l]}, n_W^{[l]}) \tag{7}$$

It is possible to create a fully connected layer with finite number of neurons. A fully connected layer takes a vector as input and returns another vector. The mathematical modelling of a fully connected layer is eq. (8).

$$z_j^{[i]} = \sum_{l=1}^{n_{i-1}} w_{j,l}^{[i]} a_l^{[i-1]} + b_j^{[i]} \tag{8}$$

Where the ϕ is activation function, w is the weight, A is the input matrix and b is the bias. It is possible to define layer $a_j^{[i]}$ as eq. (9).

$$a_j^{[i]} = \psi^{[i\}}\left(z_j^{[i]}\right) \tag{9}$$

Sometimes a neural network could be very deep and thus result may become too stable. So, it becomes too difficult to increase the accuracy of a very deep model. There are short-cut layers in ResNet architecture. These layers inject the previous layer's residuals into the network. It is possible to express the layer of a network with a residual block as eq. (10) instead of eq. (9).

$$a_j^{[i]} = \psi^{[i\}}\left(z_j^{[i]} + a_j^{[i-2]}\right) \tag{10}$$

The used ResNet-18 architecture is shown in Table 1.

Table 1. ResNet-18 Architecture.

Layer Name	Filter Size	Stride	Padding	Number of Filters	Output Feature Map Size
Input	-	-	-	-	120x240x3
Convolutional	7x7x3	2	3	64	60x120x64
BN	-	-	-	-	-
Max Pooling	3x3	2	1	-	30x60x64
Convolutional	3x3x64	1	1	64	30x60x64
Convolutional	3x3x64	1	1	64	30x60x64
Convolutional	3x3x64	1	1	64	30x60x64
Convolutional	3x3x64	1	1	64	30x60x64
Convolutional	3x3x64	2	1	128	15x30x128
Convolutional	3x3x128	1	1	128	15x30x128
Convolutional (Short cut)	1x1x64	2	0	128	15x30x128
Convolutional	3x3x128	1	1	128	15x30x128
Convolutional	3x3x128	1	1	128	15x30x128
Convolutional	3x3x128	2	1	256	8x15x256
Convolutional	3x3x256	1	1	256	8x15x256
Convolutional (Short cut)	1x1x128	2	0	256	8x15x256
Convolutional	3x3x256	1	1	256	8x15x256
Convolutional	3x3x256	1	1	256	8x15x256
Convolutional	3x3x256	2	1	512	4x8x512
Convolutional	3x3x512	1	1	512	4x8x512
Convolutional (Short cut)	1x1x256	2	0	512	4x8x512
Convolutional	3x3x512	1	1	512	4x8x512
Convolutional	3x3x512	1	1	512	4x8x512
Average Pooling	4x8	-	0	-	1x1x512

3.1.2. Quantum neural network

Using variational quantum circuits as a quantum generalization of feed-forward neural network is possible [13][16][17][18][19][20][21][22] . The definition of a quantum layer is a unitary operation which is physically realizable by a low-depth variational circuit which acts on the input |x〉 of quantum subsystems for producing the output state |y〉 as shown in eq. (11).

$$\mathrm{L}: |\mathrm{x}\rangle \ \rightarrow \ |\mathrm{y}\rangle \ = U(w)|\mathrm{x}\rangle \tag{11}$$

Where w is an array of classical variational parameters. In Quantum Neural Networks a quantum layer preserves Hilbert space dimension of the input states. A variational quantum circuit with a depth of q is a series of quantum layers which corresponds to the product of unitaries parameterized by several different weights.

In order to use classical data in a Quantum Neural Network by embedding a real vector x into a quantum state |x〉 . It can be achieved by using a variational embedding layer depending on x and applied over a reference state with eq. (12).

$$\varepsilon: \mathrm{x} \rightarrow |\mathrm{x}\rangle \ = E(x)|0\rangle \tag{12}$$

In order to parameterize single qubit rotations or single mode displacements, X is used. Y is The output vector obtained from the quantum circuit by measuring the expectation values of nq local observables where $\hat{y}$ = $[\hat{y}_{\mathbf{1}}, \hat{y}_{\mathbf{2}} \dots \hat{y}_{\boldsymbol{n_q}}]$. Measurement layer maps a quantum state to a classical vector. This layer is mathematically expressed as eq. (13).

$$\mathrm{M}\colon |\mathrm{x}\rangle \ \rightarrow \mathrm{y} = \ \langle \mathrm{x}|\hat{\mathrm{y}}|\mathrm{x}\rangle \tag{13}$$

In eq. (14) a complete quantum network with initial embedding layer and final measurement layer is expressed.

$$\mathrm{Q} = \mathrm{M} \circ \mathrm{Q} \circ \varepsilon \tag{14}$$

Where the Q is defined by the series of quantum layers which corresponds to the product of unitaries parameterized by different weights.

Dressed quantum circuits are used to connect a classical network to a quantum network. Input and output data is processed by adding classical layers at the beginning and ending of a quantum circuit. By considering the Equation (14), it is possible to express a dressed quantum circuits as (15).

$$\tilde{Q} = L_{n_q \rightarrow n_{out}} \circ \mathrm{Q} \circ L_{n_{in} \rightarrow n_q} \tag{15}$$

Where $L_{n \rightarrow n'}$ is defined as eq. (16) by considering the eqs. (8) and (9).

$$L_{n \rightarrow n'}\colon \mathrm{x} \rightarrow \mathrm{y} = \psi(wx + b) \tag{16}$$

When the e1. (16) is considered, proposed quantum circuit is not working as a co-processor to the classical processor since the main computation is done in quantum processor and only data embedding and readout processes are held in classical layers.

3.1.3. Loss function and optimizer

Cross entropy loss function is used in order to measure the performance of the proposed model. There are only 2 classes thus eq. (17) is used to calculate cross entropy.

$$\mathrm{Loss} = -[\mathrm{ylog(p)} + (1 - y)log\ (1 - p)] \tag{17}$$

Adam optimization algorithm is used for model optimization. It is an extension of classical stochastic gradient descent. Used equation to calculate weights by using Adam optimization eq. (18) is used.

$$w_t = w_{t-1} - \eta\left(\frac{\widehat{m}_t}{\sqrt{\hat{v}_t + \varepsilon}}\right) \tag{18}$$

Where $\boldsymbol{\eta}$ is the step size $\widehat{\boldsymbol{m}}$ is the w is the weight, is the bias corrected estimator of first moment and $\hat{\boldsymbol{v}}$ is the bias corrected estimator of second moment.

3.1.4. Experimental models

Two different architectures with different depths of quantum networks are developed. Architectures developed to conducts experiments are given in Table 2.

Table 2. Models used in experiments.

Classical Network	Number of Qubits in Each Quantum Layer	Depth of Quantum Network
ResNet-18	4	2
ResNet-18	4	4

3.2. Environment

When the e1. (16) is considered, proposed quantum circuit is not working as a co-processor to the classical processor since the main computation is done in quantum processor and only data embedding and readout processes are held in classical layers.

Experiments are conducted on Google Collaboratory environment. Python language is used to develop code of conduct. For building the proposed architecture Torch library is used along with the Torchvision library to load visual data. Pennylane library is used to build, train and test the quantum neural network in a simulation environment.

3.3. Dataset

In this research, a fake human face dataset with several different difficulty levels are used. The dataset is named as Real and Fake Face Detection by Computational Intelligence and Photography Lab of Yonsei University which consists a total of 2041 fake and real face [23]. 960 of them was fake and 1,345 of them was real. 30 random images are used as test set.

4. RESULTS

Several experiments are conducted in order to show the proposed quantum-classical hybrid neural network architecture's success to determine whether if an image is real or fake. In these experiments each model is trained for 20 epochs. Used hyperparameters with the 4 qubits and 2-layer model are given in Table 3.

Table 3. Hyperparameters.

Hyperparameter	Value
Learning Rate	0.0001
Batch Size	8
Gamma Learning Rate Scheduler	0.1
Q_delta	0.01

Used hyperparameters with the 4 qubits and 2-layer model are given in Table 4.

Table 4. Hyperparameters.

Hyperparameter	Value
Learning Rate	0.0004
Batch Size	8
Gamma Learning Rate Scheduler	0.1
Q_delta	0.01

Firstly, experiments on an architecture with 4 qubits with a depth of 2 quantum layers is conducted. Train and validation accuracies according to each epoch are given in Fig. 1. and train and validation loss values according to each epoch are given in Fig. 2.

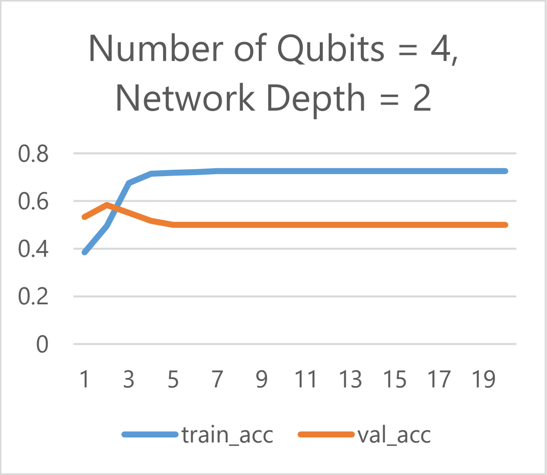

Figure 1. Train and Validation Accuracy.

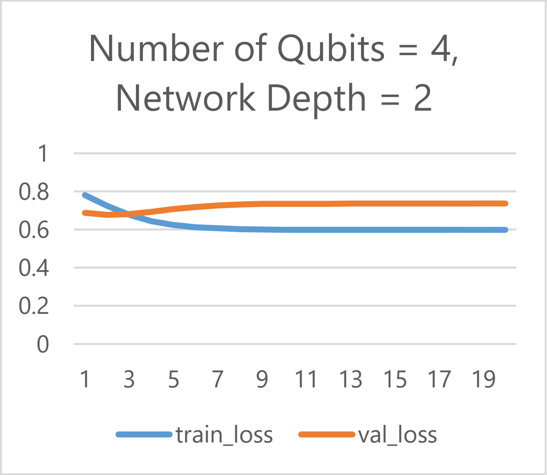

Figure 2. Train and Validation Loss.

According to the Fig. 1 and Fig. 2 it is observed that after the epoch 3 validation accuracy of the model decreases but training accuracy increases.

Secondly, experiments on an architecture with 4 qubits with a depth of 4 quantum layers is conducted. Train and validation accuracies according to each epoch are given in Fig. 3. and train and validation loss values according to each epoch are given in Fig. 4.

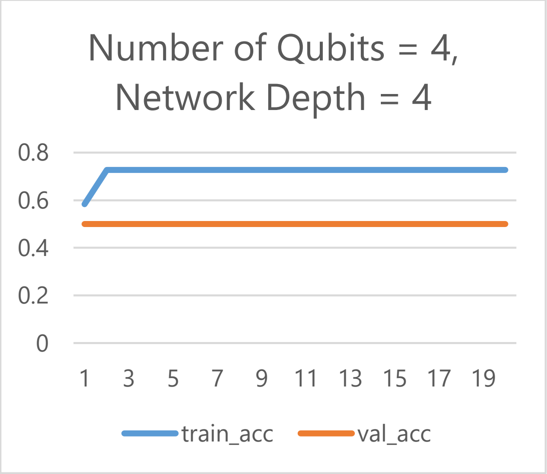

Figure 3. Train and Validation Accuracy.

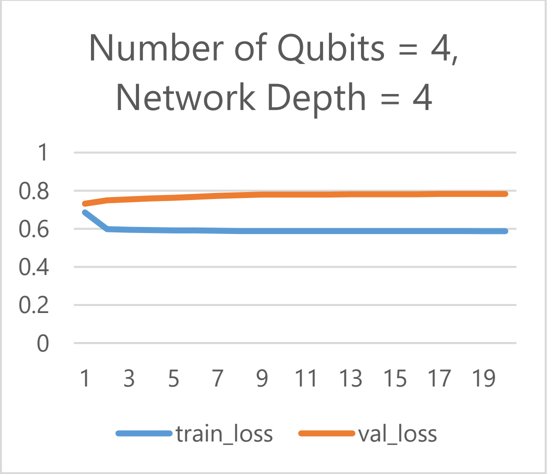

Figure 4. Train and Validation Loss.

According to the Fig. 3 and Fig. 4 it is observed that after the epoch 2 training accuracy of the model increases but no change in validation accuracy is observed.

5. DISCUSSION

Currently there are many security applications like face recognition and iris recognition based on the machine vision. Thus, it could be stated that the machine vision algorithms might be an essential part of the future of modern data security systems. Recent developments in the Generative Artificial Neural Network Models show that there is an emergent need for fake image recognition technologies to ensure the reliability of security systems since it became more easy to generate fake data. In this research we proposed to use a quantum-classical hybrid deep learning architecture with transfer learning approach to recognize whether if an image is fake or real.

This research shows that it is possible to use quantum computers in security systems. By considering the several limitations due to the ongoing research on quantum computers or quantum co-processors can be used along with classical processors to improve the reliability of security systems in the future. When quantum co-processors become realizable there will be a need to develop more complex hybrid architectures to improve the accuracy of the results.

This research could be furthered by combining different classical and quantum networks to achieve better performance. Hyperparameter optimization can solve the overfitting problem and increase the validation and test accuracies. Also training and testing the proposed model on a real quantum computer is possible.

6. CONCLUSION

This research show that usage of quantum-classical hybrid deep learning models can increase the reliability of machine vision based security systems. It can be concluded that the architecture is too deep and hyperparameters should be optimized further to prevent overfitting especially for quantum models. In this research both training and validation processes took place in simulation of a quantum computer. Training a Quantum-Classical Hybrid Neural Network architecture on a real quantum computer may slightly affect the results due to the noise but for training in simulation it is possible to conclude that quantum networks are not immune to the over-fitting problem.

REFERENCES

[1] Voulodimos, A., Doulamis, N., Doulamis A., & Protopapadakis E. Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience* 2018, 7068349.

[2] Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybernetics* 1980, 36, 193–202.

[3] Feynman, R. P. Int. J. Theor. Phys. 1982, 21, 467.

[4] Nielsen, M. A., Chuang, I. L. Quantum Computation and Quantum Information (Cambridge University, New York, 2009).

[5] Shor P. W. SIAM J. Comput. 1997, 26, 1484.

[6] Preskill, J. Quantum Computing NISQ era and Beyond, 2018, arXiv:1801.00862.

[7] Lewestein, M. Quantum Perceptrons, Journal of Modern Optics 1994, 41, (12), 2491-2501.

[8] Chrisley, R.L. Quantum learning, in: P. Pylkka¨nen, P. Pylkkö (Eds.), *New Directions in Cognitive Science, Proceedings of the International Symposium*, Saariselka¨, Finnish Artifcial Intelligence Society, Lapland, Finland, 1995.

[9] Chrisley, R.L. Learning in non-superpositional quantum neurocomputers, in: *P. Pylkka¨anen, P.Pylkkö (Eds.), Brain Mind and Physics*, IOS Press, Amsterdam, 1997, pp. 126*139.

[10] Menneer, T., Narayanan, A. Quantum inspired neural networks, *Department of Computer Science, University of Exeter*, UK, http://www.dcs.ex.ac.uk/reports/reports.html, 1995.

[11] Menneer, T., Narayanan, A. Quantum Artificial Neural Networks vs Classical Artificial Neural Networks: Experiments in Simulation, *Proceedings of the Fifth Joint Conference on Information Sciences*, 2000, vol. 1, pp. 757-759.

[12] Dallaire-Demers, P., Killoran, N. Quantum Generative Adversarial Networks, *Phys. Rev*. A, 2018, 98, 012324

[13] Killoran, N., Bromley, T.R., Arrazola, J.M., Schuld, M., Quesada, N. & Lloyd, S. Continous-Variable Quantum Neural Networks, 2018, arXiv:1806.06871.

[14] Xia, R., Kais, S. Hybrid Quantum-Classical Neural Network for Calculating Ground State Energies of Molecules, 2019, arXiv:1902.06184.

[15] Mari, A., Bromley, T.R., Izaac, J., Schuld, M., & Killoran, N. Transfer Learning in Hybrid Classical-Quantum Neural Networks, 2019, arXiv:1912.08278.

[16] Farhi, E., Neven, H., Classification with quantum neural networks on near term processors, 2018, arXiv:1802.06002.

[17] Liu, D., Ran, S., Wittek, P., Peng, C., Garc ía, R.C., Su, G., & Lewenstein, M. Machine learning by unitary ten-sor network of hierarchical tree structure. *New Journal of Physic*s, 2019, 21, (7), 073059.

[18] Perdomo-Ortiz,A., Benedetti, M., Realpe-G ómez,J., & Biswas, R. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quantum Science and Technology*, 2018, 3, (3), 030502.

[19] Peruzzo, A., McClean, J., Shad-bolt, P., Yung, M., Zhou, X., JLove, P., Aspuru-Guzik, A., & L O'brien, J. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 2014, 5, 4213.

[20] Schuld, M., Killoran, N. Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, 2019, 122, (4), 040504.

[21] Schuld, M., Bocharov, A., Svore K. M., & Wiebe, N. Circuit-centric quantum classifiers. Physical Review A, 2020, 101, (3).

[22] Sim, S., Johnson, P.D., & Aspuru-Guzik, A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2019, *2*(12), 1900070.

[23] Yonsei University Computational Intelligence and Photography Lab, Real and Fake Face Detection, 2019, Retrieved 08.02.2021 from: https://www.kaggle.com/ciplab/real-and-fake-face-detection.