

# 1.Introduction

Digital image processing (DIP) defines a collection of computational operations imposed onto the pixels of one image. The results of these operations can include but are not limited to revealing objects not initially visible in the image, detecting and distinguishing objects within the image, enhancing image quality, and analyzing patterns around objects. From DIP's introduction in 1960[1] till now, it had become a useful tool in medical Imaging, forensic Studies, textiles, material Science, Graphic Arts, etc... and as Silpa Joseph suggested "Millions of digital images are uploaded in every single minute" and "image processing techniques are of higher importance nowadays" [2], people had started seeking alternative methods to circumvent the extensive computational resources need in storing and computing with conventional computer image processing[3]. Quantum information processing (QIP), initially introduced by Paul Benioff in 1980[4] and later popularized by Richard Feynman with his groundbreaking paper on simulating physics using computers in 1982[5], became an extremely ideal solution in the future. Thanks to QIP's quantum properties natures such as entanglement, superposition, and parallel computing, this method of image processing can produce comparable results with conventional image processing methods with storage and time efficiency and lesser complexity. Most notably, a new complexity class named Bounded-Error Quantum Polynomial Time (BQP) was created to capture problems that quantum computers can solve efficiently with quantum advantages while classical computers cannot in most cases [6]. However, Quantum machines are still far from a serviceable state due to two contemporary major challenges: limitation of Qubits and Noise [7].

## 1.1 Limitation of Qubits:

Currently, well-known existing quantum machines include:

- **IBM Osprey:** This quantum processor was unveiled by IBM and has a qubit count of 433, which is more than three times the number of qubits on the previous IBM Eagle processor
- **IBM Quantum System One:** IBM's Quantum System One is one of the newest quantum computers, which was announced to have a 127-qubit processor named 'Eagle'. This machine represents a significant step towards IBM's goal of building a quantum computer with a thousand qubits.
- **Google's Sycamore Processor:** Google's quantum computer, featuring the 54-qubit Sycamore processor, achieved a milestone known as quantum supremacy by performing a specific task in 200 seconds that would take a classical supercomputer approximately 10,000 years.

However, upon examining the **Flexible Representation of Quantum Images (FRQI)**, known for utilizing a minimal number of qubits and promising potentials with best classification

accuracy compared to NEQR and other encoding methods [8], the Quantum image representation still requires  $\log_2(N)$  number of qubits, with  $N$  denoting the number of pixels in a picture. This meant that the largest input sample is  $433 \times 433 = 187489$  pixels with the most advanced IBM Osprey machine. 433 pixels is adequate for a simulating size as a simplified version of quantum algorithms, but still far from application in real life.

## 1.2 Noise in the Result:

Noise in quantum computing refers to any external or internal factors that can affect the accuracy and reliability of a quantum computer's calculations. This noise can arise from a variety of sources, such as:

- **Environmental disturbances:** Fluctuations in temperature, electromagnetic fields from devices like Wi-Fi or mobile phones, and even cosmic rays can introduce errors.[9]
- **Imperfect control signals:** The precision required to manipulate qubits is extremely high, and any deviation in the control signals can lead to incorrect qubit states.[9]
- **Unwanted interactions between qubits:** Qubits are overly sensitive to their surroundings, and interactions with neighboring qubits can cause them to change state unintentionally.[9]

These disruptions can lead to a phenomenon known as decoherence, where the quantum information stored in qubits deteriorates over time, potentially leading to the loss or randomization of data. Decoherence is particularly challenging because quantum information is extremely fragile and can be easily perturbed by noise. If the noise consistently affects the result of QIP, then the results are unusable either.

Quantum machine improvements had been consistent, and there are hopes of seeing 1000- qubits machine created by IBM [10]. However, the problem of noise in the machine is yet to be solved. In this paper, we propose a novel method in reducing noise. By creating a machine learning model that specializes in discerning images processed by classical and quantum computers, with a particular focus on identifying and rectifying noise in quantum images. This model will be trained with a training data from both classical and quantum computer processed image under supervised learning and gain the ability to indicate the confidence level of each pixel and its possible original value when fed a quantum processed image with potential noise corruption. We will evaluate this model's accuracy in compensating the loss and decoherence when image is processed through the quantum machine with signal-to-noise ratio (PSNR), structural similarity index (SSIM), and mean opinion score (MOS) standards, then focus on the model's applicability in pictures from different areas, and its cost effectiveness compared to other methods.

## **2. Typical noise after quantum image processing.**

Quantum image processing often encounters two types of noise: Salt-and-Pepper noise and Gaussian noise [11]. Salt-and-Pepper noise in images presents itself as sparsely occurring white and black pixels. It can be caused by sharp and sudden disturbances in the image signal. An effective noise reduction method for this type of noise is a median filter. In the context of quantum image processing, the Q-Mean algorithm is designed to remove the Salt-and-Pepper noise. As for Gaussian noise, it combines thermal, amplifier, and read noise into a single noise term that is independent of the unknown signal and that stays constant for specific camera settings, such as exposure time, gain or ISO, and operating temperature. Gaussian noise parameters may change when camera settings are changed. In quantum image processing, the Q-Gauss method applies a special mask to weaken the Gaussian noise pollution.

## **3. Image processes in both quantum and classical computers**

Quantum image processing provides more possibilities for image processing due to the powerful parallel computing capabilities of quantum computers. Let us now examine some image processing examples that Quantum computer can do with more efficiency than classical computers:

- Two-dimensional Image Transforms: Quantum computers have been experimentally demonstrated to perform two-dimensional image transforms, such as the Haar wavelet, Fourier, and Hadamard transforms, with an exponential speedup over their classical counterparts [12].
- Image Denoising and Edge Detection: Quantum principles have been used to improve image quality and experimental speed up in image denoising, image edge detection, and morphological operations. This is achieved using fast quantum Fourier and wavelet transform techniques in a binary quantum system [13].
- Quantum Image Encoding: Quantum image encoding techniques have been redesigned to reduce their susceptibility to errors, which can be processed more efficiently on real quantum computers [14].
- Quantum Image Representations: Efficient representations of digital images on quantum computers have been developed, which have better time complexity and quantum cost when compared with related models in the literature [15].

These operations can be the basis for selecting our dataset used for training our AI quantum denoising model. For the sake of understandability and visualization, we decided to use edge detection databases for training our model as a possible application of our AI denoising methodology.

## **4. Training the model**

### **4.1 Viable image data banks**

There are several open machine learning datasets that you can access for free. Some credible image processing datasets that are commonly used for model training are listed in the following. These public sets could also be of use in training the AI quantum denoising model.

- **BIPED (Barcelona Images for Perceptual Edge Detection) [16]:** This dataset contains 250 outdoor images of 1280×720 pixels each. These images have been carefully annotated by experts in the computer vision field. The dataset is publicly available as a benchmark for evaluating edge detection algorithms.
- **BRIND (Berkeley Reflectance, Illuminance, Normal, and Depth edges annotations) [17]:** This dataset is for edge detection based on the BSDS images.
- **SBD (Semantic Boundaries Dataset) [18]:** This dataset is a standard benchmark for contour detection. It includes 500 natural images with carefully annotated boundaries collected from multiple users. The dataset is divided into three parts: 200 for training, 100 for validation, and the rest 200 for test.
- **Cityscapes [19]:** This dataset is designed for semantic urban scene understanding. It consists of a large, diverse set of stereo video sequences recorded in streets from 50 different cities.
- **BSDS500 (Berkeley Segmentation Dataset 500) [20]:** This is a standard benchmark for contour detection. It includes 500 natural images with carefully annotated boundaries collected from multiple users. The dataset is divided into three parts: 200 for training, 100 for validation, and the rest 200 for test.

#### 4.2 Simulated Noise

Due to the current limitation of Quantum machines, simulated noise will be applied as an alternative to quantum processed image. To simulate the noise in quantum images, we use the depolarizing channel [12], which is a common model of noise in quantum computing. The depolarizing channel applies a random unitary operation to each qubit with a certain probability  $p$ , which can be interpreted as flipping the qubit to a random state. The effect of the depolarizing channel on a single qubit can be described by the following equation:

$$E(\rho) = (1 - p)\rho + \frac{p}{3}(X\rho X + Y\rho Y + Z\rho Z)$$

where  $\rho$  is the density matrix of the qubit,  $X$ ,  $Y$ , and  $Z$  are the Pauli matrices, and  $p$  is the depolarizing probability. We apply the depolarizing channel to each qubit of the quantum images with a probability  $p=0.1$ , which means that each qubit has a 10% chance of being flipped to a random state. This introduces noise to the quantum images, which can affect their quality and classification.

```
def add_noise(image_path, noise_type='gaussian', mean=0, std=1, amount=0.01):
    img = Image.open(image_path)
```

```

img_array = np.array(img)

if noise_type == 'gaussian':
    noise = np.random.normal(mean, std, img_array.shape)
    noisy_img_array = img_array + noise
elif noise_type == 'salt_pepper':
    noisy_img_array = img_array.copy()
    num_salt = np.ceil(amount * img_array.size * 0.5)
    coords = [np.random.randint(0, i - 1, int(num_salt)) for i in img_array.shape]
    noisy_img_array[coords] = 255
    num_pepper = np.ceil(amount * img_array.size * 0.5)
    coords = [np.random.randint(0, i - 1, int(num_pepper)) for i in img_array.shape]
    noisy_img_array[coords] = 0

noisy_img_array = np.clip(noisy_img_array, 0, 255).astype(np.uint8)
noisy_img = Image.fromarray(noisy_img_array)

return noisy_img

```

Note that the parameter allows “gussian” and “salt\_pepper” type noises be applied to the edge detected image. These are both common noises encountered in quantum computing, and thus should to a fair degree represent the noises encountered in a real quantum machine.

The following example is taken from the BIPEDv2 [21] image dataset:



- (1) RGB\_003: Original image in the dataset
- (2) RGB\_003: Edge detected with classical computer & methods
- (3) RGB\_003: Simulated noises of Quantum computer

## 5 Methodology

In this section, we describe the methodology of our proposed approach to reduce noise in quantum images. Our approach consists of three main components: a machine learning model, a training data generation process, and a noise reduction algorithm.

**5.1 Machine learning model:** We use a convolutional neural network (CNN) as our machine learning model to classify and correct quantum images. A CNN is a type of deep learning model that can

learn features from images by applying multiple layers of convolutional filters, pooling operations, and nonlinear activations. Our CNN has the following architecture:

- **Input layer:** The input layer takes a quantum image as a two-dimensional array of complex numbers, representing the amplitudes of the qubits. The input size is  $n \times n$ , where  $n$  is the number of pixels in one dimension of the image.
- **Convolutional layer 1:** The first convolutional layer applies 32 filters of size  $3 \times 3$  with a stride of 1 and a padding of 1 to the input, resulting in 32 feature maps of size  $n \times n$ . We use the rectified linear unit (ReLU) as the activation function for this layer.
- **Pooling layer 1:** The first pooling layer performs a max pooling operation with a filter size of  $2 \times 2$  and a stride of 2, reducing the size of each feature map by half to  $n/2 \times n/2$ .
- **Convolutional layer 2:** The second convolutional layer applies 64 filters of size  $3 \times 3$  with a stride of 1 and a padding of 1 to the output of the first pooling layer, resulting in 64 feature maps of size  $n/2 \times n/2$ . We use the ReLU as the activation function for this layer.
- **Pooling layer 2:** The second pooling layer performs a max pooling operation with a filter size of  $2 \times 2$  and a stride of 2, reducing the size of each feature map by half to  $n/4 \times n/4$ .
- **Fully connected layer 1:** The fully connected layer 1 flattens the output of the second pooling layer into a one-dimensional vector of size  $64n^2/16$ , and applies a linear transformation with a weight matrix of size  $256 \times 64n^2/16$  and a bias vector of size 256, followed by a ReLU activation function, resulting in a vector of size 256.
- **Fully connected layer 2:** The fully connected layer 2 applies a linear transformation with a weight matrix of size  $2 \times 256$  and a bias vector of size 2, followed by a softmax activation function, resulting in a vector of size 2. This vector represents the probability distribution over the two classes: classical or quantum.
- **Output layer:** The output layer returns the predicted class label (0 for classical, 1 for quantum) and the confidence score (the probability of the predicted class) for the input image.

The code snippet below shows the implementation of our CNN model in Python using the PyTorch framework:

```
import torch

import torch.nn as nn

import torch.nn.functional as F

class CNN(nn.Module):

    def __init__(self, n):

        super(CNN, self).__init__()

        self.conv1 = nn.Conv2d(1, 32, 3, padding=1)

        self.pool1 = nn.MaxPool2d(2, 2)

        self.conv2 = nn.Conv2d(32, 64, 3, padding=1)

        self.pool2 = nn.MaxPool2d(2, 2)
```

```

self.fc1 = nn.Linear(64 * n * n // 16, 256)

self.fc2 = nn.Linear(256, 2)

def forward(self, x):
    x = x.view(-1, 1, n, n) # reshape the input to a 4D tensor
    x = F.relu(self.conv1(x)) # apply convolutional layer 1
    x = self.pool1(x) # apply pooling layer 1
    x = F.relu(self.conv2(x)) # apply convolutional layer 2
    x = self.pool2(x) # apply pooling layer 2
    x = x.view(-1, 64 * n * n // 16) # flatten the output
    x = F.relu(self.fc1(x)) # apply fully connected layer 1
    x = F.softmax(self.fc2(x), dim=1) # apply fully connected layer 2 and softmax

    return x

```

We train our CNN model using the cross-entropy loss function and the Adam optimizer with a learning rate of 0.001 and a batch size of 32. We use the PyTorch built-in functions *nn.CrossEntropyLoss* and *torch.optim.Adam* for this purpose. We train our model for 10 epochs, and evaluate its performance on the validation set after each epoch. We use the accuracy metric to measure the performance of our model, which is defined as the ratio of correctly classified images to the total number of images.

**5.2 Training data:** To train our CNN model, we need a dataset of images that are processed by both classical and quantum methods, and labeled with their corresponding class (0 for classical, 1 for quantum). We generate this dataset as follows:

- Classical images: We use a set of grayscale images from the MNIST dataset [11], which contains 60,000 images of handwritten digits from 0 to 9. Each image has a size of 28×28 pixels, and each pixel has an intensity value between 0 and 255. We label these images as 0 (classical).
- Quantum images: We use the Flexible Representation of Quantum Images (FRQI) [8] to encode the classical images into quantum images. FRQI is a quantum image representation method that uses a minimal number of qubits and has promising potentials for quantum image processing. FRQI encodes an image of size  $n \times n$  pixels into  $\log_2 n$  qubits, where each qubit represents a row or a column of the image. The amplitude of each qubit state corresponds to the intensity value of the pixel in that row or column. For example, the image below:

!image can be encoded into two qubits as follows:

$$|\psi\rangle = \frac{1}{4}(|00\rangle + 2|01\rangle + 3|10\rangle + 4|11\rangle)$$

where the coefficients 1, 2, 3, and 4 represent the intensity values of the pixels in the top-left, top-right, bottom-left, and bottom-right corners, respectively. We use the FRQI encoding method to convert the classical images from the MNIST dataset into quantum images, and label them as 1 (quantum).

**5.2.1 Data split:** We split the dataset of classical and quantum images into three subsets: training, validation, and test. We use 80% of the data for training, 10% for validation, and 10% for test. We shuffle the data before splitting to ensure a balanced distribution of classes and digits in each subset.

**5.3 Noise reduction algorithm:** The goal of this algorithm is to use the machine learning model's output to estimate the confidence level and the original value of each pixel in a noisy quantum image, and then apply a threshold to filter out the noise and restore the image quality. The algorithm is as follows:

1. Input: A noisy quantum image  $Q$  of size  $N \times N$ , encoded by FRQI method, and the trained machine learning model  $M$ .
2. Output: A noise-reduced quantum image  $Q'$  of size  $N \times N$ , encoded by FRQI method.
3. Initialize an empty quantum image  $Q'$  of size  $N \times N$ .
4. For each pixel  $p$  in  $Q$ , do the following:
  - Feed  $p$  to the machine learning model  $M$  and get the output  $o$ , which is a vector of two probabilities:  $o=[P_c, P_q]$ , where  $P_c$  is the probability that  $p$  is a classical image pixel, and  $P_q$  is the probability that  $p$  is a quantum image pixel.
  - Calculate the confidence level  $c$  of  $p$  as the difference between  $P_q$  and  $P_c$ :  $c=P_q - P_c$ .
  - If  $c$  is greater than or equal to a predefined threshold  $T$ , then  $p$  is considered as a quantum image pixel and its original value  $v$  is estimated by the machine learning model  $M$  as the most likely quantum state among the possible states of  $p$ .
  - If  $c$  is less than  $T$ , then  $p$  is considered as a noise pixel and its original value  $v$  is set to zero, which corresponds to the black color in FRQI encoding.
  - Assign  $v$  to the corresponding pixel in  $Q'$ .
5. Return  $Q'$  as the noise-reduced quantum image.



### 3.3.2 Effectiveness analysis (metrics)

The effectiveness of the model can be evaluated using metrics such as the Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Mean Squared Error (MSE).

For more detailed instructions on training the model, you might want to refer to the research paper titled "Quantum Image Denoising: A Framework via Boltzmann Machines, QUBO, and Quantum Annealing"<sup>12</sup>. This paper investigates a framework for binary image denoising via restricted Boltzmann machines (RBMs) that introduces a denoising objective in quadratic unconstrained binary optimization (QUBO) form and is well-suited for quantum annealing<sup>12</sup>. The denoising objective is attained by balancing the distribution learned by a trained RBM with a penalty term for derivations from the noisy image<sup>12</sup>. The paper also provides detailed instructions on training Boltzmann Machines<sup>45</sup>.

## 4 Application of the Proposed Denoising Model

The proposed denoising model aims to improve the quality and reliability of quantum image processing (QIP) by using a machine learning model to identify and rectify noise in quantum images. This model can have various applications in different domains that require image processing, such as:

**4.1 Medical imaging:** QIP can be used to enhance the contrast, resolution, and segmentation of medical images, such as X-rays, MRI, CT, and ultrasound. However, noise can degrade the quality and accuracy of these images, which can affect the diagnosis and treatment of diseases. The proposed denoising model can help to reduce the noise and improve the clarity and fidelity of medical images, which can benefit both patients and doctors.

**4.2 Forensic studies:** QIP can be used to extract and analyze information from digital images, such as fingerprints, face recognition, handwriting, and document verification. However, noise can obscure or distort the features and details of these images, which can affect the identification and authentication of suspects and evidence. The proposed denoising model can help to remove the noise and enhance the visibility and recognition of forensic images, which can assist in criminal investigations and justice.

**4.3 Textiles and material science:** QIP can be used to measure and characterize the properties and quality of textiles and materials, such as fiber, yarn, fabric, and composite. However, noise can interfere with the measurement and evaluation of these properties, such as color, texture, pattern, and defect. The proposed denoising model can help to eliminate the noise and improve the measurement and evaluation of textile and material images, which can support the design and production of textiles and materials.

1. Azriel Rosenfeld, *Picture Processing by Computer*, New York: Academic Press, 1969
2. Joseph, Silpa. "IMAGE PROCESSING TECHNIQUES AND ITS APPLICATIONS: AN OVERVIEW." *IJARIE*, vol. 3, no. 8, 2017, pp. 8745-87501
3. Dendukuri, A., & Luu, K. (2018). Image Processing in Quantum Computers ArXiv preprint arXiv:1812.11042 12. <https://arxiv.org/pdf/1812.11042.pdf>
4. P. Benioff - The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines, *J. Stat. Phys.*, 22 (1980), 563-591.
5. R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6-7, pp. 467–488, 1982.
6. Roy, Shibdas. "PSPACE BQP." arXiv.org, no. 2301.10557, 24 Jan. 2023, [arxiv.org/pdf/2301.10557.pdf](https://arxiv.org/pdf/2301.10557.pdf)
7. Kwak, Yunseok, et al. "Quantum Neural Networks: Concepts, Applications, and Challenges." *arXiv.org*, arXiv:2108.01468v1 [quant-ph], 2 Aug 2021,
8. T. Nguyen, I. Paik, H. Sagawa and T. C. Thang, "Quantum Machine Learning with Quantum Image Representations," 2022 IEEE International Conference on Quantum Computing and Engineering (QCE), Broomfield, CO, USA, 2022, pp. 851-854, doi: 10.1109/QCE53715.2022.00142.
9. Konstantinos Georgopoulos, Clive Emary, and Paolo Zuliani. "Modelling and Simulating the Noisy Behaviour of Near-term Quantum Computers." *Phys. Rev. A* 104, 062432 (2021). arXiv:2101.02109 [quant-ph]. doi: 10.1103/PhysRevA.104.062432.
10. "IBM Releases First-Ever 1,000-Qubit Quantum Chip." *Scientific American*, 5 Dec. 2023, [www.scientificamerican.com/article/ibm-releases-first-ever-1-000-qubit-quantum-chip/](https://www.scientificamerican.com/article/ibm-releases-first-ever-1-000-qubit-quantum-chip/). Accessed 13 Dec. 2023.
11. Phillip Kerger and Ryoji Miyazaki. ["Quantum Image Denoising: A Framework via Boltzmann Machines, QUBO, and Quantum Annealing." arXiv preprint arXiv:2307.06542v3 \(2023\).](https://arxiv.org/abs/2307.06542v3)
12. Yao, Xi-Wei; Wang, Hengyan; Liao, Zeyang; Chen, Ming-Cheng; Pan, Jian; et al. (11 September 2017). "Quantum Image Processing and Its Application to Edge Detection: Theory and Experiment". *Physical Review X*. **7** (3):

13. Chakraborty, S., Dey, L. (2023). Quantum Computing in Image Processing. In: Computing for Data Analysis: Theory and Practices. Data-Intensive Research. Springer, Singapore. [https://doi.org/10.1007/978-981-19-8004-6\\_8](https://doi.org/10.1007/978-981-19-8004-6_8)
14. Geng, Alexander, Moghiseh, Ali, Redenbach, Claudia and Schladitz, Katja. "Quantum image processing on real superconducting and trapped-ion based quantum computers" *tm - Technisches Messen*, vol. 90, no. 7-8, 2023, pp. 445-454. <https://doi.org/10.1515/teme-2023-0008>
15. Nasr, N., Younes, A. & Elsayed, A. Efficient representations of digital images on quantum computers. *Multimed Tools Appl* **80**, 34019–34034 (2021). <https://doi.org/10.1007/s11042-021-11355-4>
16. Soria, Xavier, Riba, Edgar, and Sappa, Angel. "Dense Extreme Inception Network: Towards a Robust CNN Model for Edge Detection." The IEEE Winter Conference on Applications of Computer Vision (WACV '20), 2020.
17. Soria, Xavier, Pomboza-Junez, Gonzalo, and Sappa, Angel Domingo. "LDC: Lightweight Dense CNN for Edge Detection." IEEE Access, vol. 10, 2022, pp. 68281-68290.
18. Martin, D., Fowlkes, C., Tal, D., and Malik, J. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics." Proc. 8th Int'l Conf. Computer Vision, vol. 2, July 2001, pp. 416-423.
19. Cordts, Marius, Omran, Mohamed, Ramos, Sebastian, Rehfeld, Timo, Enzweiler, Markus, Benenson, Rodrigo, Franke, Uwe, Roth, Stefan, and Schiele, Bernt. "The Cityscapes Dataset for Semantic Urban Scene Understanding." Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
20. Martin, D., Fowlkes, C., Tal, D., and Malik, J. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics." Proc. 8th Int'l Conf. Computer Vision, vol. 2, July 2001, pp. 416-423.
21. Soria, Xavier, et al. "Dense Extreme Inception Network for Edge Detection." Pattern Recognition, vol. 139, 2023, pp. 109461, doi:10.1016/j.patcog.2023.109461.
22. 12, 2022. Available: <https://ieeexplore.ieee.org/document/9971386>. [Accessed: Dec. 13, 2023][12]G. Hu, W. Jing, W. Guo, and M. Abbas, "Opposition-based learning boosted orca predation algorithm with dimension learning: a case study of multi-degree reduction for NURBS curves," Journal of Computational Design and Engineering, vol. 10, no. 2, pp. 722–757, Mar. 2023, doi: <https://doi.org/10.1093/jcde/qwad017>. Available: <https://academic.oup.com/jcde/article/10/2/722/7069334>.

