Classification of the Polish Handwritten Letters by the use of Quantum Convolutional Neural Network

Joanna Wiśniewska[1 [0000−0002−2119−3329]] and Marek Sawerwain[2 [0000−0001−8468−2456]]

[1] Institute of Information Systems, Faculty of Cybernetics, Military University of Technology, Gen. S. Kaliskiego 2, Warszawa 00-908, Poland
JWisniewska@wat.edu.pl

[2] Institute of Control & Computation Engineering, University of Zielona Góra, Licealna 9, Zielona Góra 65-417, Poland
M.Sawerwain@issi.uz.zgora.pl

Abstract. The problem of data classification realized by the convolutional networks, although successfully implemented using classical methods, is also important in the area of quantum networks and is subject to continuous development and research. In this work, we present an example of classification for a set of higher dimensionality than the currently used solutions based on the MNIST or FASHION databases. Additionally, we show that working on raw data not transformed by, for example, PCA reduction or other advanced classical pre-processing techniques, very high classification quality can be achieved also without using any hybrid techniques. In the discussed solution, classification is performed by checking whether the obtained final state, or more precisely the probability distribution of the basis states superposition, is consistent with the appropriate state representing the given label. This comparison can be carried out using basic techniques such as Kullback–Leibler divergence or SWAP-Test, especially if we want the classification process to be realized solely in the quantum computation model without using any post-processing with classical techniques.

Keywords: quantum machine learning · variational quantum algorithm · quantum convolutional neural network · numerical experiment

1 Introduction

It can be observed that in a past few years, the great development took place in the fields of Artificial Intelligence (AI) and quantum computing. Today, AI is extremely popular and wanted. There are great hopes and expectations regarding AI's possibilities and universality, which probably will become more realistic over time. Anyway, there are some use cases in which AI's methods are very efficient, e.g. pattern recognition, natural language processing. On the other hand, we can observe a development of quantum computers which can realize some

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

computations much faster than their classical counterparts. It is very tempting to join two mentioned solutions and propose quantum AI.

The history of artificial neural networks begins in 1943 with a neuron's model constructed by McCulloch and Pitts [12]. With time, solutions based on mentioned model are becoming more complex – there are more neurons and links between them, including feedback, and groups of specialized neurons. Nowadays, one of the most popular type of artificial networks are the Convolutional Neural Networks (CNNs) [7]. It is a method from a group termed as deep learning because of many layers of neurons. CNNs are useful in image processing and computer vision. In this work, we show the efficiency of Quantum CNNs (QCNNs) [3], [8] in recognition of letters, more precisely, signs of Polish alphabet which originates from the Latin alphabet with some modifications, e.g. ą, ę, ó, ś, ż. In general, QCNNs are an innovative approach to data processing, combining the capabilities of quantum computers with advanced techniques used in classical neural networks [2]. Such computational architectures offer the potential to significantly improve analytical processes in areas such as visual data analysis, optimization, or modelling physical phenomena.

It should also be noted that the development of quantum convolutional networks [3], [11] and the upcoming availability of Noisy Intermediate-Scale Quantum (NISQ) is important in the context of realization of physical quantum machines. In the article, we also pay attention to the fact that the designed QCNN has an architecture based on gates operating on neighbouring qubits [9]. The obtained simulation results (for more classes than currently discussed in the literature [8], [17]) allow for effective classification of data based on such a network. As it seems, it is also worth emphasizing that the proposed solution allows us to point out the following contributions: the method proposed in the paper does not involve pre-processing of data, e.g. by PCA [6] to limit the data dimensionality. Particular observations are encoded directly using the amplitude technique by the use of superposition principle which requires only the logarithmic amount of the quantum information units when we encode classical data, and still allows obtaining high classification accuracy. Additionally, post-processing of data does not require the use of a hybrid approach, where, for example, an additional classical network is used to analyze the obtained quantum state. The solution presented here is the last step of the classification process because the probability distribution comparison procedure can be implemented using the SWAP-Test which can be also realized as a quantum circuit. We also use the whole register to encode output label of the single input image as a suitable probability distribution, instead of the using a single qubit to encode the final label of the input image. The architecture of quantum convolution network can be based on the typical quantum gates, like, NOT gate and controlled R_X, R_Y, R_Z gates which can be placed as adjacent gates.

Organization of the presented chapter is as follows: in Sec. 1.1, we present notations, symbols, abbreviations, and definitions which are used in the chapter. In Sec. 2, we outline the structure and some technical aspects of data set which is used in presented research. We also discuss a problem of encoding classical data

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

of individual images as quantum data. Sec. 3 presents quantum convolutional neural network expressed in terms of quantum circuits. Sec. 4 is devoted to an analysis of performed numerical experiment and a discussion about achieved accuracy. Conclusions are contained in Sec. 5. Acknowledgments and References are the last parts of the chapter.

1.1 Notations, symbols, and abbreviations

Before we start a presentation of our solution's proposal for classification of handwritten letters and digits and an introduction of necessary definitions for QCNNs, we summarize in Table 1 notations, symbols, abbreviations and acronyms used in the chapter.

Table 1. Some symbols, notations, sets and functions used in the paper

Notation	Description
N_Q, N_I, N_L, N_{QL}	number of qubits, total number of images, number of labels (all integers), and N_{QL} represents the number of qubits used to encode labels
X, X_j	set of classical data (letters and digits) and single image of j-th char
Y, $\hat{Y}$, Y_j, $\hat{Y}_j$	true labels set, the predicted labels set and the j-th label for true and predicted sets
L, L_j	set of labels and the j-th label of a particular sign
$QC_k(\theta^k)$, $QP(\theta^l)$	respectively, k-th quantum convolutional layer and polling layer with individual parameters
$\mathcal{F}$	Fidelity value between two quantum states
$R_z^{(i)}$	rotation gate Z applied to the i-th qubit
$CR_x^{(i,j)}$	controlled rotation gate X applied to the (i+1)-th qubit (where the i-th qubit is a controlling one)
$\mathbb{R}$, $\mathbb{C}$, $\mathbb{N}$	sets of real, complex, and integer numbers,
$\partial E(\mathbb{C}^d)$	set of pure states on $\mathbb{C}^d$ i.e. complex space for d qubits
$1 \ldots n$	means the sequence of $1, 2, 3, \ldots, n$

2 PolLettDS – Polish handwritten data set

The MNIST database [10] is an example of handwritten digits which is broadly utilized in evaluating efficiency of many different techniques termed as Machine Learning (ML). This database is very popular, mainly, by its size which is 60,000 digits in the training set and 10,000 digits in the test set. This fact makes it a very useful tool for assessing the effectiveness of newly developed methods. However, the MNIST database does not contain any letters and its resolution is low, i.e. 24×24 pixels.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

In this work, we present a freshly gained set of handwritten letters PolLettDS [16] which also includes signs characteristic for the Polish language. Mentioned set is going to be regularly expanded with new data. At the time of writing this article, we have 4160 images of digits, lowercase, and uppercase letters. In relation to the MNIST database, the images are bigger because there are 64×64 pixels in the grayscale.

Table 2 shows the basic information about the data set used in the process of classification. It should be emphasized that each sign is associated with a label which clearly defines the sign's meaning. Each set of handwritten signs contains 80 probes (ten digits, lowercase, and capital letters). Polish alphabet is made of 32 letters but we have also added letters: q, v, x, therefore, in total, we consider 35 letters.

Summarizing, our data set contains 4160 images and 80 labels including ten digits, lowercase, and uppercase letters. The number qubits N_Q needed to encode data results from the image of each sign. The total number of images N_I, the number of labels N_L, and the number of qubits N_Q utilized to encode labels are:

$$N_I = 4160, \quad N_L = 80, N_Q = \log_2 64 \cdot 64 = 12. \tag{1}$$

Further comments on the method of representing the image X_j as a state of quantum register is given in Sec. 2.1.

Definition 1. *The whole set* PolLettDS *is marked as X and a single j-th image as X_j. Analogously, the set of all labels is Y and each j-th image has a label $Y_j \in L$. The set of labels predicted by the classifier are denoted as $\hat{Y}$ and its elements as $\hat{Y}_j$.*

	Number	Resolution	Type of Pixel
Number of sets:	52	–	
Digits in each set:	10	64×64 pixels	encoded in grayscale
Lowercase letters in each set:	35		
Uppercase letters in each set:	35		
Number of labels in each set:	$N_L = 80$	–	–
Total:	$(10 + 35 + 35) \cdot 52 = 4160$ chars and digits ($N_I = 4160$)		

Table 2. The classification process, described in this work, is performed on 52 sets of signs (each set contains digits, lower- and uppercase letters)

The half of set's signs was entered with the use of a digital drawing tablet and a graphic program in which each sign was entered into a grid cell. This allowed for an easy process of acquisition and further images processing. The latter half of signs was handwritten on an A4 sheet of paper and scanned in 300 DPI resolution. Next, the images were scaled to the resolution 64×64 pixels

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

and converted to the grayscale. In Fig. 1, we can see exemplary images of signs and also both grids used to enter characters – the digital one and the prepared paper sheet.

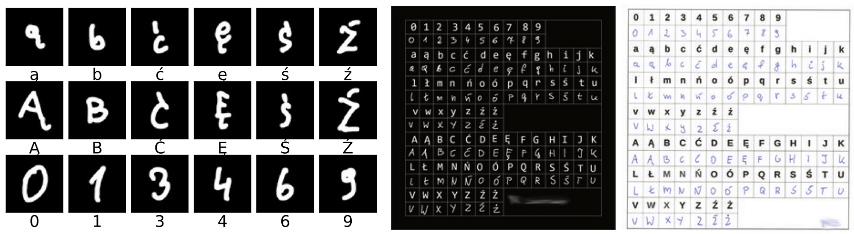

Fig. 1. At the left, we can see exemplary images from the PolLettDS (the first row contains lowercase letters, the second – uppercase letters, and the last row shows digits). The black grids represent the digital grids and, on the right, the paper form is presented

2.1 Quantum representation of data set

A representation of classical data – in this case, grayscale images with resolution 64×64 pixels – is, naturally, possible to express as a state of a quantum register. We utilize here the superposition phenomenon which is present in a quantum computations model. For each image X_j the corresponding quantum state $|\psi_j\rangle \in \mathbb{C}^{2^{N_Q}}$ (more formally we encode image X_j as pure state $|\psi_j\rangle \in \partial E(\mathbb{C}^{N_Q})$ in the set of all states belong to the space $\mathbb{C}^{N_Q}$ represents N_Q qubits system) may be denoted with the use of amplitude coding approach [18]:

$$X_j \longmapsto |\psi_j\rangle = \sum_{i=0}^{2^N-1} \alpha_i |i\rangle, \tag{2}$$

where the brightness level of particular pixel from single image X_j is described by probability amplitude α_i related to a basis state $|i\rangle$ which represents pixel coordinates. This way of data encoding requires a transformation of pixels values from the two-dimensional image X_j to a column vector and the normalization of amplitudes.

Remark 1. It should be emphasized that this procedure do not allow to encode a completely black image with N qubits. If all pixels are black, so the level of brightness would be zero for each probability amplitude and the quantum state would not be normalized. We assume that in such cases one more qubit is needed and an empty image will be expressed as a basis state with the extra qubit equal $|1\rangle$ and other qubits set up to $|0\rangle$.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

Because of the applied method of coding images as quantum states and the Remark 1, the number og needed qubits for an image sized $w \times h$ is: $N_Q = \log_2(w \cdot h) + 1$, where w stands for the image's width and h is the height in pixels. For the sake of simplicity, we may assume that the product $w \cdot h$ equals $2^{N_Q - 1}$. This shows that one image from the PolLettDS requires only logarithmic number of qubits to be encoded as probability amplitudes of a quantum state.

The labels from the Y set are consistent to examples shown in Fig. 1, i.e. digits codes are the same as labels, e.g. the label of digit 5 is 5. However, letters also have assigned numerical values, e.g. "j" is 22, capital letter "A" is 50. Therefore, all labels are integers $Y_j \in \mathbb{N}$. Naturally, one can directly assign a numerical value to a label in a quantum register $|\psi_{Y_j}\rangle$: $|\psi_{Y_j}\rangle = |0110010\rangle$, where binary number 00110010 is decimal 50, i.e. capital letter "A". The following bits describe the states of seven qubits encoding the label. We expect that the classifier recreates the final quantum state for each image according to this rule. Such solution would be susceptible to a noise and bit-flip errors [15], which may be reduced with quantum correction codes but it would increase the number of needed qubits. However, another approach can be applied. The label may be encoded by the use of quantum superposition of all register's qubits:

$$|\psi_{L_k}\rangle = \frac{1}{\sqrt{N_{\text{spc}}}} \sum_{i=0}^{N_{\text{spc}}} \alpha_{i+spc} |i + spc\rangle, \tag{3}$$

where $N_{\text{spc}} = \lfloor 2^{N_Q}/\text{spc} \rfloor$ and spc is a range (offset) between amplitudes, e.g. if spc $= 128$ then for "A" encoded in state $|\psi_{L_k}\rangle$ the following probability amplitudes with indexes 50, 178, 306, ..., 4018 will be active. Now, a change of probability amplitudes values by, for example, amplitude-damping channel [5], may still result with a correctly indicated label. On the other hand, the bit-flip errors may influence the register's state but the offset allows skipping qubits particularly susceptible to errors of this type in case of hardware implementation. Label classification also requires comparing two probability distributions, i.e. the ideal one encoding a given label with the obtained state after the implementation of the quantum classification circuit, what we discuss in the next section and in Sec. 4.

Another important assumption is the orthogonality of states encoding labels:

$$\forall_{k,l \in L} \quad \langle \psi_{L_k} | \psi_{L_l} \rangle = 0, \tag{4}$$

what causes that the quantum states of the labels are uniquely related to the images X_j. However, in case of lack of clarity, the Fidelity measure computed for two states or the SWAP-test [1] allows checking the degree of similarity of the obtained distribution with the label distributions $|\psi_{Y_j}\rangle$.

3 Quantum convolutional circuit for classification

A Quantum Convolutional Neural Network (QCNN), depicted in Fig. 2, applied to the classification of the data set is a direct counterpart of the classical CNN.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

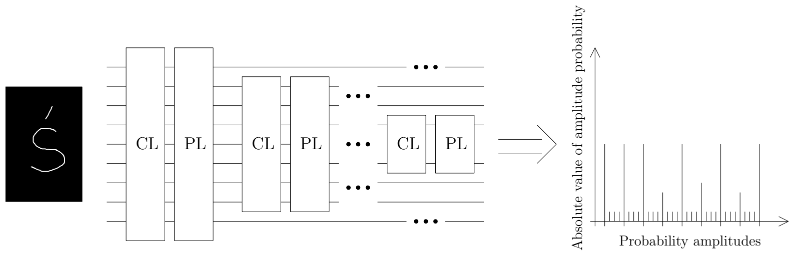

Fig. 2. The general scheme of utilized QCNN (CL – convolutional layer, PL – polling layer). For the image X_j (in the picture Polish letter "ś"), we expect that the network will transform the image into a given probability distribution representing the label associated with the given image. In this case, classification is based on comparing the obtained distribution with the target distribution corresponding to a specific letter or number

For example, a set of quantum gates parameters θ is equivalent to the set of weights and biases in CNN. Particular quantum gates, mainly rotation gates, act like a non-linear activation functions. The input state and its processing by layers of quantum gates relates to the feed-forward neural network's architecture. However, we should remember the main difference between quantum and classical circuits – in a quantum system neither any information can be copied nor qubits may be directly rejected. The quantum computation includes the whole register because of the high probability of quantum entanglement's presence between qubits. Adding qubits during the computation is not possible. Discarding qubits can be realized by the quantum measurement which irreversibly changes the register's state and destroys the entanglement (completely or partially). This means that some qubits may be omitted but it is a serious operation influencing the whole register. Even if some qubits are omitted during the processing, they will physically remain a part of the quantum system. In our case, we expect that the QCNN, shown in Fig. 2, returns a quantum state which is a superposition of basis states. This superposition will be the same or similar to previously assumed state and its probability distribution. The probability distribution obtained from a series of measurements can be compared at the post-processing level or at the quantum circuit level using the so-called SWAP-Test [1], in order to compare a suitably prepared quantum state with the label representing the QCNN output.

The circuit, more detailed, applied to the classification of signs from the PolLettDS, is characterized by a structure shown in Fig. 3. In case of our data set (section 2.1), according to the adopted way of information coding, the input of the circuit is constructed of twelve qubits what allows loading an image X_j to the register. Then, the register's state is transformed by convolutional QC and polling QP layers. In pursuance of Fig. 3, the first pair of QC and QP layers affects all utilized qubits, but next pairs of layers include smaller numbers of qubits. The PolLettDS contains eighty different labels and the last QCNN's layer includes seven qubits what enables coding 2^7 probability amplitudes – fully

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

covering the number of labels. The number 2^7 is sufficient to train the QCNN with images X_j and to ensure the final state with the adequate probability distribution representing the label Y_j.

The realization of the QCNN with n layers is based on unitary operators QC_k, QP_k in each k-th layer and parameters describing convolutional $\theta_{k,c}$ and polling $\theta_{k,p}$ subcircuits:

$$\mathrm{QCNN}(\theta) = \mathrm{QC}_n(\theta_{n1}) \cdot \mathrm{QP}_n(\theta_{n2}) \cdot \ldots \cdot \mathrm{QC}_2(\theta_{22}) \cdot QP_2(\theta_{21}) \cdot QC_1(\theta_{12}) \cdot QP_1(\theta_{11}). \tag{5}$$

In general, we define the convolutional circuit as:

$$QC_n(\theta_{n1}) = \left(\prod_{i=0}^{N_Q} R_y^{(i)}(\theta_{n_1}^k) R_z^{(i)}(\theta_{n_1}^l) \right) \left(\prod_{i=0}^{N_Q-1} CR_x^{(i,i+1)}(\theta_{n_1}^m) \right), \tag{6}$$

where $k, l, m \in \{1 \ldots N_Q\}$. $R(i)_z(\theta_{n_1}^k)$ means that a particular gate is applied to the i-th qubit. $CR_x^{(i,i+1)}(\theta_{n_1}^m)$ is a controlled rotation gate where i is the controlling qubit and $i+1$ points out the controlled qubit.

The definition of the polling layer is:

$$QP_n(\theta_{n2}) = \prod_{i=0}^{N_Q-1} CR_z^{(i,i+1)}(\theta_{n_1}^k) X^{(i)} CR_z^{(i,i+1)}(\theta_{n_1}^l), \tag{7}$$

with the markings as in the convolutional layers.

After the unitary operation QCNN(θ) the whole quantum state has to be measured and the obtained probability distribution will indicate the new values of parameters θ. The construction of subcircuits QC and QP is very important because they introduce the entanglement into the whole register. However, in next layers the number of qubits is smaller, so the changes are performed only for the last qubits which should encode the final probability distribution for the labels. The subcircuits QC and QP are constructed of typical quantum gates, i.e. Pauli gates, 1-qubit rotation gates, controlled negation gates and controlled rotation gates. Moreover, we expect that the gate operations will be used for adjacent qubits what facilitates a hardware implementation of such designed circuits.

In Fig. 3, except the whole circuit's structure, we can see the schemes of subcircuits QC and QP. The quantum register's properties, especially superposition and unitary operator's utilization, significantly limits the number of parameters that we control in the case of the designed convolutional network.

For the analyzed data set X, we utilize N_Q qubits in the classifying circuit and the labels are encoded by N_{QL} qubits, the number of QCNNs parameters is:

$$N_\theta = \sum_{k=N_Q-N_{QL}+2}^{N_Q-1} \left(3_c \cdot k + 2_p \cdot k \right), \tag{8}$$

where $(N_{QL}+2) < (N_Q - 1)$ and 3_c stands for the number of parameterized gates in the convolutional layer and 2_p, analogously, for the polling layer. For the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

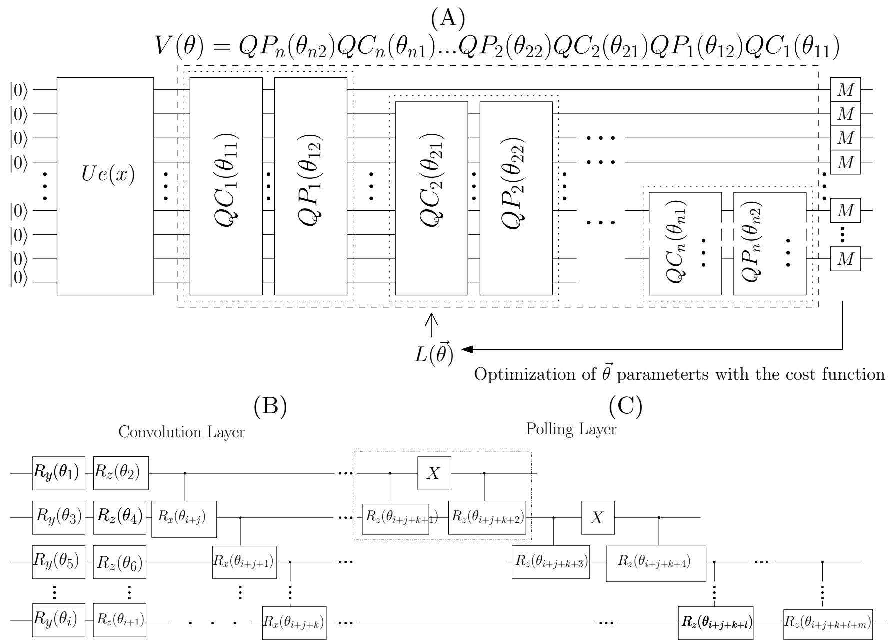

Fig. 3. General scheme (A) of a QCNN with several layers of convolutional (QC) and polling (QP) sub-circuits. The first operation $Ue(X_j)$ performs the encoding of classical image X_j to the quantum state. The number of layer depends on the size of the final output state. In our case, the input state requires 12 qubits, but the number of labels (80 labels) is covered by the eight qubits, so only five pairs of QC and QP layers are used. The last layer acts only on seven qubits. Part (B) shows the convolutional circuit where after the rotation gates R_x and R_y, the controlled gates CR_x are placed. In part (C), the polling section is depicted – therein, the circuit realizes controlled rotation CR_z and X operation for the subsequent qubits

case considered in the article, we have three gates for the convolutional phase and two gates for the polling phase for each qubit considered in the k-th layer. That gives a total number of 225 parameters.

The operation of the network as a classifier can be described in a direct way. For the initial state $|\psi_j\rangle$ representing an image X_j, the QCNN realizes a unitary operation:

$$|\psi_j^{\hat{Y}_j}\rangle = \mathrm{QCNN}(\theta_j)|\psi_j\rangle. \tag{9}$$

We expect that the state $|\psi_j^{\hat{Y}_j}\rangle$ with parameters θ_j represents the predicted label $\hat{Y}_j$. The state $|\psi_j^{\hat{Y}_j}\rangle$ have to be measured to reveal the final probability distribution of the basis states that construct this state.

Selection of parameters θ is based on the loss function definition which utilizes the Fidelity measure between the final state $|\psi_j^{\hat{Y}_j}\rangle$ and the state which describes

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

the correct label of an image X_j. The loss function is:

$$L(X,Y) = \frac{1}{N_I} \sum_{j=0}^{N_I-1} 1 - \mathcal{F}\Big(|\psi_j^{\hat{Y}_j}\rangle, |Y_j\rangle) \Big), \tag{10}$$

and the aim is to minimize its value by a proper selection of parameters θ:

$$\min_{\theta} \quad L(X,Y). \tag{11}$$

In the numerical experiment, described in Sec. 4, the minimization is based on the COBYLA method [14].

3.1 Classification process as quantum circuit

Naturally, the QCNN and the probability distribution obtained at its output is sufficient to realize the classification task. However, the analysis of probability distribution is carried out at the classical level, e.g. by the Kullback–Leibler divergence [4]. On the other hand, the classification process by comparing the probability distribution can also be done by using the SWAP-Test, using the measurement of one qubit only. A scheme of this type of approach is presented in Fig. 4.

By measuring one qubit, after applying the SWAP-Test, we can expect the state $|0\rangle$ with probability $\frac{1}{2}$, which means that the compared states presenting the labels are not similar to each other, whereas the probability of measuring the state $|1\rangle$ close to one determines a very high consistency of states, i.e. the probability that the label encoded by the state $|\psi_j^{\hat{Y}_j}\rangle$ is similar or identical to the label $|L_k\rangle$.

The presented solution does not dismiss us from repeating the computations but limits the number of measured qubits to practically one, which in the case of NISQ machines allows reducing errors related to decoherence and noise, improving the accuracy of the results while maintaining computational efficiency.

4 Experiment result and accuracy evaluation

The numerical experiment was performed using the NVIDIA CUDA-Q [13] package. The work environment included a computer with AMD Ryzen 9 7950X processor and NVIDIA RTX 6000 ADA graphics card, and also the WSL environment for Windows 11. However, it should be emphasized that the training process within the NVIDIA CUDA-Q environment, despite the use of a modern graphics card, still takes a relatively long time, e.g. tuning parameters only for a single letter using the COBYLA optimizer takes from 200 to 250 seconds performing 2048 optimizer's steps.

The learning process, despite the above difficulties, was characterized by a good convergence, as shown in Fig. 5 for the capital Polish letter "Ż". However, as it may be seen, the COBYLA optimizer for 225 parameters naturally achieved

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

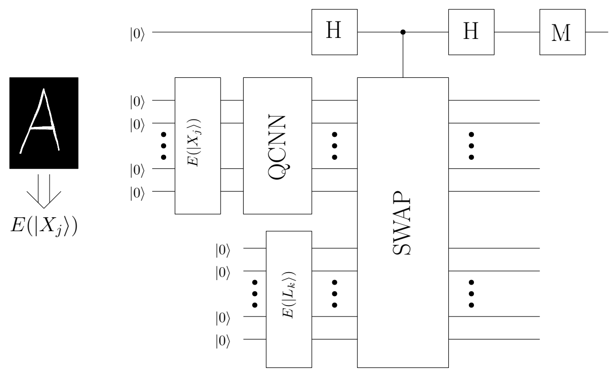

Fig. 4. General scheme of using QCNN for classification. After inputting image X_j, encoded into quantum state by circuit $E(|X_j\rangle)$, and executing the QCNN circuit, the obtained quantum state, i.e. predicted label, can be checked for compliance with labels from set L using the SWAP-Test

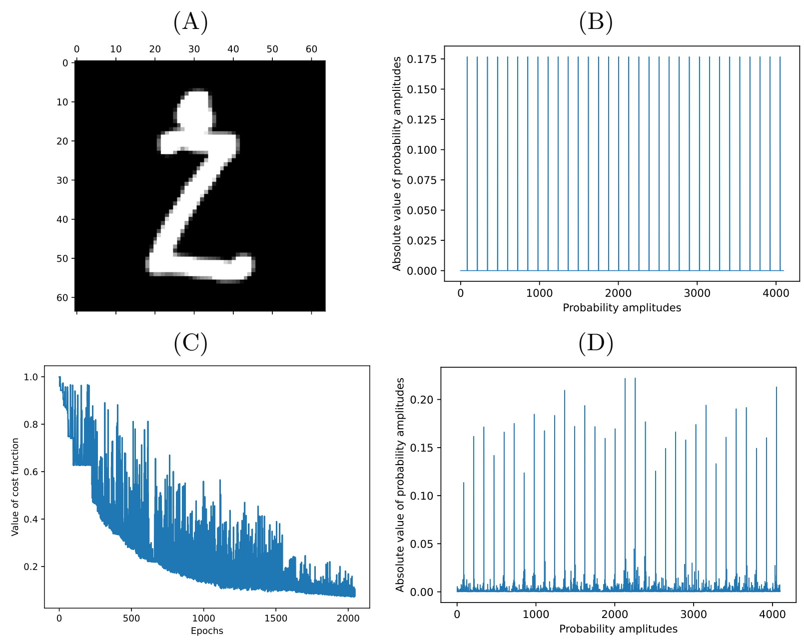

Fig. 5. The process of tuning the values of the θ parameters for an exemplary letter "Ż" of the Polish alphabet. Illustration (A) shows the image of the letter, (B) is the desired description of the quantum state where individual pins according to the formula Eq. 3 represent the label for the letter "Ż". (C) depicts the value of the cost function during the parameters tuning process, and (D) is the obtained distribution of probability amplitudes which, although noisy, is characterized by a high Fidelity value of ≈ 0.92793, relatively to the distribution (B)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

the assumed value of the objective function without any problems, i.e. the Fidelity value reached the value of ≈ 0.93, but the learning process in this case is characterized by a high variability, and even remaining in a local minimum for a certain number of steps, which can be seen in the aforementioned figure.

It should be emphasized that Fidelity values above 0.9 already suggest very high classification quality. Due to the long computation time, but nevertheless without loss of generality, it is possible to perform a classification quality test using, for example, arbitrarily selected ten characters: 0, 5, 9, A, ą, Ł, o, ś, X, Ż.

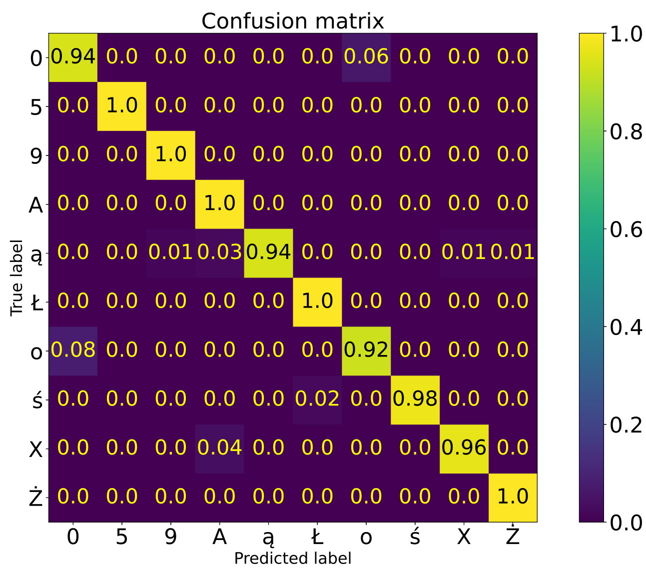

Fig. 6. The matrix confusion for ten signs: 0, 5, 9, A, ą, Ł, o, ś, X, Ż. High accuracy was achieved, however, characters similar to each other in handwriting, i.e. 0 and the lowercase letter o, as can be seen, have reduced quality of classification in relation to other classes

In each case, we have 80 letters of each type in the data set. Therefore, we can select the parameters of the QCNN network for a specific digit or letter. In our case, this means preparing ten sets of values for the parameters θ associated with the selected characters and training the network so that the loss function for a specific character is as low as possible. Having a set of parameters θ_{L_k} for the selected label, we are able to classify the image X_j, if the Fidelity between the quantum state $|\psi_j^{\hat{Y}_j}\rangle$ and $|\psi_{L_k}\rangle$ exceeds the value 0.9.

The classification results are collected in Fig. 6. The obtained results confirm the high efficiency but also bring out typical problems of classification, e.g. zero and the lowercase letter "o", which in the case of handwriting are sometimes written in a very similar way. This will also apply to the uppercase "O". Similarly,

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

a reduced classification quality can be encountered for the letters: "Z", "Ź", "Ż". Where in the case of less careful handwriting, these signs can also pose a problem for correct classification.

The source code, as well as data for the discussed implementation of the classification task, can be found in [16]. It should be emphasized that the use of the NVIDIA CUDA Q package allows for easy creation of appropriate circuits for the discussed task. This package creates so-called computational kernels, whose task is to implement a quantum circuit. Fig. 7 shows the beginning of the kernel implementing the QCNN network. Starting the procedure, in order to obtain a quantum state that can be further processed, has been very simplified:

```
rslt_qstate = cudaq.get_state( qae_step_01_qcnn_classification,
  theta_parameters )
```

The obtained object represents a quantum state that can be further processed using other computational kernels.

```
@cudaq.kernel
def qae_step_01_qcnn_classification( angles: List[float] ):
    register = cudaq.qvector( input_qstate_00 )
    aidx = 0
    # Begin of Layer 1, Convolutional layer
    for idx in range( number_of_qubits-1 ):
        ry ( angles[ aidx ], register[ idx ] )
        aidx = aidx + 1
    for idx in range( number_of_qubits-1 ):
        rz ( angles[ aidx ], register[ idx ] )
        aidx = aidx + 1
    for idx in range( number_of_qubits-1 ):
        rx.ctrl( angles[ aidx ], register[idx], register[idx+1] )
        aidx = aidx + 1
    # Pooling layer
    for idx in range( number_of_qubits-1 ):
        rz.ctrl(angles[ aidx ], register[idx], register[idx+1])
        aidx = aidx + 1
        x(register[idx])
        rx.ctrl(angles[ aidx ], register[idx], register[idx+1])
        aidx = aidx + 1
    # End of Layer 1

... the rest of the code ...
```

Fig. 7. A piece of the Python code utilizing NVIDIA CUDA Q technology, which implements the first convolutional and polling layers for the circuit realizing the QCNN from Fig. 3

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

5 Summary

The article presents the application of a QCNN to handwriting recognition. The task of the network was to categorize a set of signs to 80 classes, where individual images have a resolution of 64 by 64 pixels. High quality of character prediction was obtained, with the QCNN of a relatively small number of parameters, i.e. 225. This shows the significant advantage of quantum solutions in this area of application. Although the learning process, in aspect of the numerical simulation, still constitutes a significant burden, the prediction process itself can already be carried out and simulated very efficiently. The adopted method of building a circuit based on the neighborhood of quantum gates, and using only basic types of gates, also allows for further experiments on current quantum hardware of the NISQ type. Naturally, it is possible to improve the quality of recognition by extrapolating the circuit structure. The completed classification task also showed the usefulness of the PolLettDS set, as a benchmark, despite the relatively small number of characters. Naturally, it is planned to systematically expand it with further examples of handwriting. Another important problem is the noise presence with NISQ hardware but the elastic structure of QCNN should allow us to overcome this problem, which is the next important goal of our investigation. The main tool for simulation in this article was NVIDIA CUDA Q package but other quantum framework can be also used to create application with the PolLettDS. Finally, it should be also emphasized that the data set will be supplied with new probes of handwritten letters and digits.

Acknowledgments. This work was financed by Military University of Technology under research project UGB 531-000023-W500-22 and by a subsidy for research projects in Technical Computer Science and Telecommunication discipline in University of Zielona Góra for year 2025.

Disclosure of Interests. All authors declare that they have no conflicts of interest.

References

1. Aïmeur, E., Brassard, G., Gambs, S.: Machine learning in a quantum world. In: Lamontagne, L., Marchand, M. (eds.) Advances in Artificial Intelligence. pp. 431–442. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
2. Chen, S.Y.C., Wei, T.C., Zhang, C., Yu, H., Yoo, S.: Quantum convolutional neural networks for high energy physics data analysis. Phys. Rev. Res. **4**, 013231 (Mar 2022). https://doi.org/10.1103/PhysRevResearch.4.013231
3. Cong, I., Choi, S., Lukin, M.D.: Quantum convolutional neural networks. Nature Physics **15**, 1273–1278 (2019). https://doi.org/10.1038/s41567-019-0648-8
4. Csiszar, I.: *I*-Divergence Geometry of Probability Distributions and Minimization Problems. The Annals of Probability **3**(1), 146 – 158 (1975). https://doi.org/10.1214/aop/1176996454
5. Fletcher, A.S., Shor, P.W., Win, M.Z.: Channel-adapted quantum error correction for the amplitude damping channel. IEEE Transactions on Information Theory **54**(12), 5705–5718 (2008)

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17

6. Gong, L.H., Pei, J.J., Zhang, T.F., Zhou, N.R.: Quantum convolutional neural network based on variational quantum circuits. Optics Communications **550**, 129993 (2024). https://doi.org/https://doi.org/10.1016/j.optcom.2023.129993
7. Habibi Aghdam, H., Jahani Heravi, E.: Guide to Convolutional Neural Networks (2017)
8. Henderson, M., Shakya, S., Pradhan, S., Cook, T.: Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Machine Intelligence **2**(1), 2 (Feb 2020). https://doi.org/10.1007/s42484-020-00012-y
9. Kandala, A., Mezzacapo, A., Temme, K., Takita, M., Brink, M., Chow, J.M., Gambetta, J.M.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. Nature **549**(7671), 242–246 (Sep 2017). https://doi.org/10.1038/nature23879
10. LeCun, Y.: The MNIST database of handwritten digits (1998), https://yann.lecun.com/exdb/mnist/, last accessed 2025/01/13
11. Maouaki, W.E., Marchisio, A., Said, T., Bennai, M., Shafique, M.: AdvQuNN: A Methodology for Analyzing the Adversarial Robustness of Quanvolutional Neural Networks . In: 2024 IEEE International Conference on Quantum Software (QSW). pp. 175–181. IEEE Computer Society, Los Alamitos, CA, USA (Jul 2024). https://doi.org/10.1109/QSW62656.2024.00033
12. McCulloch, W., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics **5**, 115–133 (1943)
13. NVIDIA: Documentation of NVIDIA CUDA Q (2025), https://developer.nvidia.com/cuda-q
14. Powell, M.J.D.: A view of algorithms for optimization without derivatives. Technical report damtp, Cambridge University (2007)
15. Riste, D., Poletto, S., Huang, M.Z., Bruno, A., Vesterinen, V., Saira, O.P., DiCarlo, L.: Detecting bit-flip errors in a logical qubit using stabilizer measurements. Nature communications **6**(1), 6983 (2015)
16. Sawerwain, M.: Github repository for PolLettDS data set (2025), https://github.com/qMSUZ/PolLettDS
17. Shi, S., Wang, Z., Li, J., Li, Y., Shang, R., Zhong, G., Gu, Y.: Quantum convolutional neural networks for multiclass image classification. Quantum Information Processing **23**(5), 189 (May 2024). https://doi.org/10.1007/s11128-024-04360-7
18. Weigold, M., Barzen, J., Leymann, F., Salm, M.: Expanding data encoding patterns for quantum algorithms. In: 2021 IEEE 18$^{\text{th}}$ International Conference on Software Architecture Companion (ICSA-C). pp. 95–101. IEEE (2021). https://doi.org/10.1109/ICSA-C52384.2021.00025

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97570-7_17