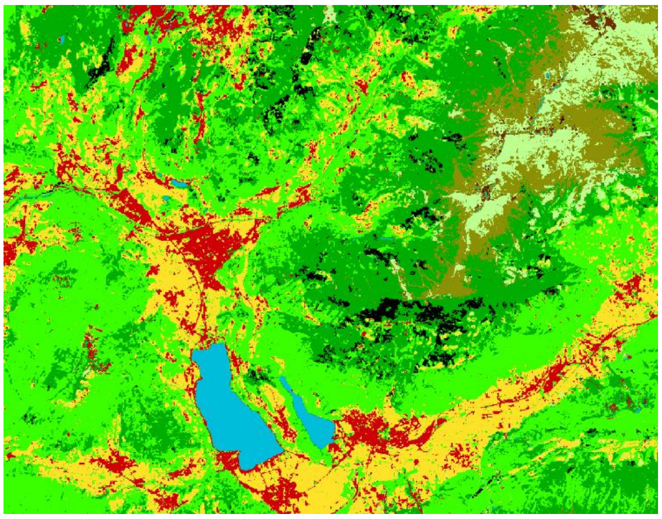
Quantum Computing for Earth Observation (QC4EO) Study

Summary of UC3 Optical Satellite Data Analysis



🟥 urban 🟩 grass 🟨 crop 🟫 bareland 🟩 broadleaves 🟩 conifers 🟩 shrub 🟦 water

Optical satellite data analysis is indispensable for extracting valuable insights from acquired data, enabling the comprehensive study and enhanced understanding of terrestrial and oceanic processes, as well as Earth's dynamic systems. This encompasses crucial functions such as land cover monitoring, environmental assessment, precision weather forecasting, disaster preparedness and response, cutting-edge atmospheric research, and its application in diverse scientific and urban planning contexts. The QC4EO study focuses on the process of Land Use and Land Cover (LULC) classification, which aims at interpreting the information obtained by satellite data to create classification maps of the investigated scene. Classification maps are thematic products of great importance for many EO applications, such as monitoring deforestation, resource management, agriculture, and the study of the impact of climate change. This use case is addressed using quantum kernel methods. Kernel methods are a well-established framework in machine learning and they can be understood as a two-step methodology. The first step involves mapping the data from the original input space into a higher-dimensional kernel feature space through a nonlinear function. The second step focuses on solving a linear problem within this transformed kernel space. These methods enable the design and interpretation of learning algorithms in the kernel space, which is nonlinearly related to the input space, effectively combining statistics and geometry. Importantly, they provide solutions with the desirable property of uniqueness, often requiring only a few sets of free parameters to ensure proper algorithm functioning. The mapping into quantum information is carried out by applying a data-dependent unitary transformation to an initial reference state. The circuit responsible for encoding, often referred to as the feature map, should be computationally challenging to simulate using classical methods; otherwise, it may not yield a quantum advantage. The quantum kernel function between two data points is obtained by taking the modulo square of the dot product between the quantum states obtained by encoding the corresponding feature vectors. Such a quantity cannot be directly calculated but must be estimated through a sampling procedure by performing measurements on the quantum state. The number of shots used to estimate each kernel function evaluation scales quadratically with the inverse of the additive error that one wants to achieve. To perform the training of the learning algorithm, it is necessary to evaluate the kernel function across all possible data pairs, and therefore, the number of such evaluations scales quadratically with the number of data points. The quantum kernel can then be used by classical supervised learning algorithms such as Support Vector Machines (SVM) and Gaussian Processes (GP). These algorithms can be implemented on various quantum hardware platforms such as superconducting and trapped ion hardware. The number of qubits needed for this approach is strictly related to the number of features of data points, requiring a few hundred qubits for data with a high number of features (e.g., hyperspectral images, time series of data, etc.). The main obstacle for this approach is the computational time needed to calculate the kernel function evaluations between the data points. Access to quantum hardware resistant to errors is also important to obtain a good estimation of the kernel function. The QC4EO study anticipates that a full-size problem might be possible to solve within a 15-year timeframe. However, even though there are feature maps that are conjectured to be hard to simulate classically, it is not yet clear how the quantum kernel implementation might provide an advantage compared to classical solutions.

In the most common implementation of a quantum kernel algorithm, each feature from the data vector is encoded in a single qubit of the quantum register. Such encoding is carried out by applying some parametrized quantum gates whose parameters depend on the feature value, for instance by applying qubit rotation with an angle depending on the feature value. The encoding can be applied multiple times to increase the model's expressivity (data-reuploading algorithms). The number of required qubits is strictly related to the number of features in the data and amounts to about several hundred qubits for high-dimensionality hyperspectral data. The required qubits connectivity in the quantum hardware depends on the encoding strategy that is used: when using the so-called "full-entanglement" strategy a CNOT gate is applied to each possible pair of qubits, thus requiring full connectivity, however, there are
other possible entanglement strategy in which the CNOT gates are applied to a small subset of qubits pairs. To achieve a higher model's expressivity, the encoding should ideally be repeated several times thus increasing the circuit's depth. Such a higher depth will entail a higher number of gates and therefore error correction will be needed to ensure the correctness of the computation. Moreover, to estimate the kernel function value through a sampling procedure several run of the quantum circuits are needed. Such sampling procedure must be carried out for each possible data point pair (thus scaling quadratically with the dataset size). A low gate-operation time is therefore advisable in order to maintain the execution time low for a large problem with many image pixels. The two main quantum platform candidates for this application are superconducting qubits and ion-based quantum computers. Superconducting qubit hardware, such as those produced by IBM and IQM, provide, in general, a high number of physical qubits and a low gate time execution. The companies working with superconducting qubits have also plans to increase the number and the quality of qubits in the future: for instance, IBM is scheduled to provide a quantum computer with thousands of logical qubits in 2033. Ion trap quantum computers are another candidate for building quantum computing hardware: in general, they offer a higher connectivity and

higher error fidelity when compared to some superconducting implementations, at the expense of a lower number of qubits. One of the main companies developing ion-trap quantum computing hardware is IONQ which currently provides quantum computer with a few tens qubit and it is planning to reach 1024 qubits in 2028, according to their technological roadmap.

The current estimated times for the execution of a 2-qubits gate on superconducting and ion-trap hardware are 533 ns and 50 ms, respectively. By considering those values, it is possible to get an estimate for the execution time of a problem instance on quantum hardware by multiplying the number of 2-qubits gates by their corresponding execution times. For this use case each quantum kernel instance must be calculated for each possible pair of data points, i.e., for on N(N-1)/2 values, with N being the number of pixels. For each of those values, the number of 2-qubits gates depends on the specific encoding structure, for example, when using a full entanglement scheme, it amounts to n(n-1) CNOT gates, with n being the number of qubits. The feature map can then be repeated by an arbitrary number of times L (data-reuploading models). Finally, for each kernel value to be estimated the circuit must run for a number of times that scales quadratically with the inverse of the average additive error that one wants to achieve. By taking these notions into consideration, it is possible to get an estimation of the number of CNOT gates and therefore the execution time.

	Problem size	**Hardware requirements**	**Timeline**
			Up to 15 years
Minimum-size problem	Learning problem with ~1.000 training samples	Digital hardware (superconducting, ion-trap): number of qubits fixed, equal to number of features (up to ~100), number of gates scales linearly/quadratically according to the entanglement scheme. Execution time has to be reasonably short.	The problem can be implemented, but the feasibility depends on the gate times. The total time can be computed as described above. Currently, the execution time for a sufficient accuracy bound is prohibitive. As no roadmaps are available, no estimation can be provided for the future.
Full-size problem	Learning problem with ~10.000 training samples	Digital hardware (superconducting, ion-trap): number of qubits fixed, equal to number of features (up to ~100), number of gates scales linearly/quadratically according to the entanglement scheme. Execution time has to be reasonably short.	A time factor of 100 with respect to the minimum-size problem must be accounted.