Quantum Annealing for Single Image Super-Resolution

Han Yao Choong
ETH Zürich
Switzerland
hchoong@student.ethz.ch

Suryansh Kumar*
ETH Zürich
Switzerland
sukumar@ethz.ch

Luc Van Gool
ETH Zürich
Switzerland
vangool@ethz.ch

Abstract

This paper proposes a quantum computing-based algorithm to solve the single image super-resolution (SISR) problem. One of the well-known classical approaches for SISR relies on the well-established patch-wise sparse modeling of the problem. Yet, this field's current state of affairs is that deep neural networks (DNNs) have demonstrated far superior results than traditional approaches. Nevertheless, quantum computing is expected to become increasingly prominent for machine learning problems soon. As a result, in this work, we take the privilege to perform an early exploration of applying a quantum computing algorithm to this important image enhancement problem, i.e., SISR. Among the two paradigms of quantum computing, namely universal gate quantum computing and adiabatic quantum computing (AQC), the latter has been successfully applied to practical computer vision problems, in which quantum parallelism has been exploited to solve combinatorial optimization efficiently. This work demonstrates formulating quantum SISR as a sparse coding optimization problem, which is solved using quantum annealers accessed via the D-Wave Leap platform. The proposed AQC-based algorithm is demonstrated to achieve improved speed-up over a classical analog while maintaining comparable SISR accuracy[1].

1. Introduction

The problem of single image super-resolution (SISR) aims at the reconstruction of a credible and visually adequate high-resolution (HR) image from its low-resolution (LR) image representation. Unfortunately, this problem is inherently ill-posed, as each independent pixel could be mapped to many HR image pixels depending upon the given upsampling factor. Nevertheless, a reasonable solution to SISR is critical to many real-world applications concern-
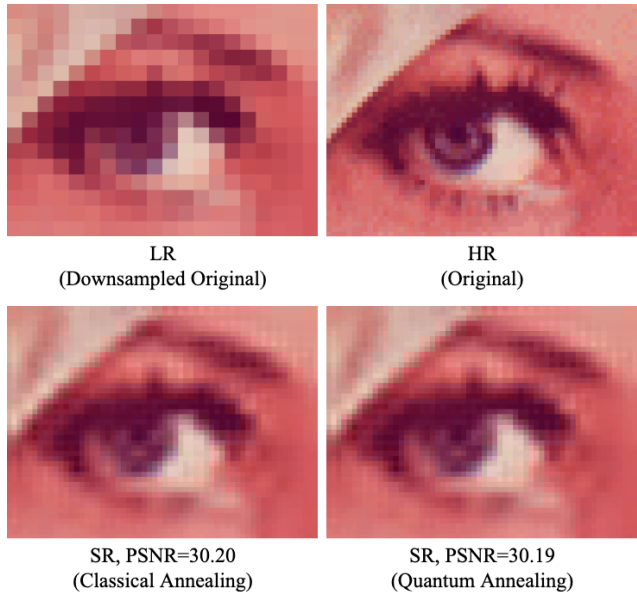

Figure 1. Results on a cropped region of the Lenna image with a magnification factor of 3. **Top row.** Ground-truth LR, HR. **Bottom row.** *Left:* Result of the Classical Annealing approach showing PSNR value of 30.20 with an inference time of 658.36s. *Right:* Result of our Quantum Annealing approach, showing similar PSNR accuracy of 30.19 with an inference time of 89.60s.

ing image data, such as remote sensing, forensics, medical imaging, and many more. One of the most attractive properties of SISR for these applications is that it can generate desirable HR images that may not be realizable by onboard camera hardware.

In recent years, SISR methods based on deep learning (DL) have become prominent, demonstrating DL's suitability on different metrics such as PSNR accuracy, inference speed, memory footprint, and generalization [13, 21, 37, 38, 41]. In this paper, our goal is not to argue against the prevailing view on the performance that could be achieved on the available computing machines via deep-networks but to explore the suitability of quantum computing for

---

*Corresponding Author (k.sur46@gmail.com)

[1]Due to the limited availability of quantum computing resources for academic research, it is difficult to **fully** demonstrate and document the true potential of quantum computing on the inference time and accuracy.

SISR. Such a motivation seems possible now since very recently functional quantum computers are available for experimentation and research—although in a limited capacity [1, 25, 28]. At present, in the domain of quantum computing, quantum processing units (QPUs) carrying out adiabatic quantum computing (AQC) have reached a critical phase in terms of the number of qubits per QPU, making application to practical computer vision problems a reality [4, 14].

The margins of this draft are limited to discuss all the advantages of quantum computing, and readers may refer to [31, 32] for deeper insights. The motivation of our work is to investigate the use of current adiabatic quantum computers and the benefits they can have compared to a well-known optimization formulation in SISR. That brings us to formulate the quantum SISR problem as quadratic unconstrained binary optimization (QUBO) problem, posing a challenge to map it to the quantum computer. Specifically, our approach relies on a popular assumption in sparse coding *i.e.*, a high-resolution image can be represented as a sparse linear combination of atoms from a dictionary of high-resolution patches sampled from training images [39].

Consequently, our contribution is demonstrating a practical SISR approach using AQC. To this end, a QUBO formulation for sparse coding is developed and adapted for D-Wave Advantage quantum annealers. The resulting algorithm can carry out approximate sparse coding and is benchmarked against a classical sparse coding approach [39] and a state-of-the-art deep learning SISR model [22]. As an early work in the synthesis of AQC for SISR, the approach proposed in this paper can serve as a basis for the further development of algorithms to exploit quantum parallelism better. As mentioned before, our proposed AQC algorithm achieves improved speed-up over a classical approach with similar accuracy when tested on popular datasets.

2. Related Work

Since our approach is one of the early approaches for solving SISR on quantum computing approaches, we discuss some popular SISR methods and existing quantum computing-based computer vision methods in this section.

(a) **Single Image Super-Resolution (SISR).** As is well-known, HR images are related to their LR counterparts by a non-analytic downsampling operation with generally permanent information loss [34]. As such, with HR images being the causal source of observed LR images, the inference of HR from LR images casts SISR to be an inverse problem that approximates the inverse of the non-analytic downsampling, with the possibility of utilizing assumptions or prior knowledge about the HR-to-LR transformation model [19, 39]. Furthermore, SISR has the property of being ill-posed, as a single LR image observation can have multiple HR mapping as valid non-unique solutions [19].

For decades, SISR has been an active research area with different approaches to solve it reliably and efficiently [29, 34, 39]. Among early approaches are observation model-based methods [12, 15], interpolation [7, 16, 33] and sparse representation learning [39, 40]. More recently, DL approaches have set the new state-of-the-art in SISR regarding image reconstruction accuracy, inference speed, and model generalization, thereby enabling real-time inference at resolution as high as 4K for video applications [10, 20, 36]. As alluded to above, our work focuses on adapting a traditional SISR method to the quantum setting; thus, in this paper, we skip a detailed discussion on state-of-the-art DL-based approaches. Interested readers may refer to [3, 6] for a comprehensive overview of the recent state-of-the-art in SISR.

(b) **Quantum computing and its current state in computer vision.** At present, central processing units (CPUs) and quantum processing units (QPUs) are used concurrently, forming a hybrid quantum-classical algorithm. The advantage offered by quantum computing comes at one or more particular critical steps in the algorithm design consisting of problems that are conventionally expensive to solve with only CPUs, such as integer factorization [17], graph cut [8, 35], unstructured search [18], and other important combinatorial optimization problems [26]. To that end, applying quantum computing to such problems can result in commendable speedup over CPU computing by exploiting quantum parallelism *i.e.*, the ability to perform operations on exponentially many superimposed memory states simultaneously [30].

Within modern quantum computing, two paradigms exist to solve different problems amenable to quantum parallelism: universal gate quantum computing and adiabatic quantum computing (AQC). In this paper, we focus on the adiabatic quantum computing paradigm. AQC is a computational method that uses the quantum annealing process to solve a quadratic unconstrained binary optimization (QUBO) problem of the form

$$\operatorname*{argmin}_{\mathbf{z}} \mathbf{z}^T \mathbf{Q} \mathbf{z} + \mathbf{b}^T \mathbf{z}, \tag{1}$$

where, $\mathbf{Q} \in \mathbb{R}^{N \times N}$ contains quadratic QUBO coefficients, $\mathbf{b} \in \mathbb{R}^N$ contains linear QUBO coefficients, and $\mathbf{z} \in \{0, 1\}^N$ contains the binary variables for optimization.

As QUBO is a combinatorial optimization problem that AQC can solve, formulating a computer vision problem as a hybrid classical-quantum algorithm with a QUBO step may enable the computation of the problem to be accelerated compared to a purely classical computing algorithm. In the same spirit, several works in computer vision research have used state-of-the-art AQC hardware containing thousands of qubits to demonstrate the application of AQC to problems such as point correspondence [14], shape matching [4], object tracking [43], and others [9, 27, 42]. This work pro-

poses a SISR algorithm developed on D-Wave Advantage quantum annealer QPUs available via the D-Wave Leap interface. Contrary to other quantum computer vision applications, our work is an initial attempt to solve SISR where the previous approaches may not be suitable—reasons will become clear in the later part of the paper.

In the following section, we provide a brief overview of the concepts on which our approach is developed.

3. Preliminaries

Sparse Signal Representation. The application of quantum computing to SISR requires the development of an algorithm that contains a step with a problem amenable to solving by QPUs. Current state-of-the-art approaches to SISR based on deep neural networks do not readily satisfy this criterion and as such a suitable SISR problem formulation without involving neural networks design is required. One popular and outstanding SISR approach that could be adapted for AQC is sparse signal representation of SISR problem via l_1 minimization of the optimization variable in a cost function. To that end, we derive SISR as a sparse l_1 quantum optimization problem, much of which is inspired from the work of Yang et al. [39]. In our formulation, sparse coding is employed as the basic computational problem to be adapted for quantum computing.

In sparse coding, the output $\mathbf{x} \in \mathbb{R}^M$ is recovered as a linear combination of $\boldsymbol{\alpha} \in \mathbb{R}^N$—usually sparse, and a overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{M \times N}$ in the form $\mathbf{x} = \mathbf{D}\boldsymbol{\alpha}$. Here, M is the problem dimensionality, N is the number of atoms in the dictionary, while dictionary overcompleteness entails $N > M$. Yet, the use of QPUs to carry out binary optimization results to a binary sparse coding formulation, where $\boldsymbol{\alpha}$ consists of binary variables *i.e.*, $\boldsymbol{\alpha} \in \{0, 1\}^N$ rather than continous floating values. In SISR via sparse representation, the basic take is to train a dictionary pair $(\mathbf{D}_l, \mathbf{D}_h)$ from image patches and then to use sparse approximation patch-wise to optimize $\boldsymbol{\alpha}$ w.r.t. $\mathbf{D}_l$ and the LR patches, subsequently generating pixel intensity information for each HR patch from $\mathbf{D}_h$ and the optimized $\boldsymbol{\alpha}$.

4. Proposed Approach

4.1. Problem Formulation

SISR via sparse signal representation that uses classical l_1 optimization is popularly formulated as

$$\boldsymbol{\alpha}^* = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \|\mathbf{D}_l\boldsymbol{\alpha} - \tilde{\mathbf{y}}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1. \quad (2)$$

To find the sparse coefficients for recovery of HR image patches by $\mathbf{x} = \mathbf{D}_h\boldsymbol{\alpha}^*$, Eq.(2) is the main optimization problem solved at inference time. Instead of using conventional algorithms such as LARS [11], an AQC-based algorithm could be used. To adapt the optimization problem in

Eq.(2) to a format that facilitates AQC, a QUBO formulation involving binary vectors to be optimized is necessary. To this end, a binary mask $\mathbf{m}$ is introduced, leading to a new optimization objective with the binary mask $\mathbf{m}$

$$\mathcal{L} = \|\mathbf{D}_l(\mathbf{m} \odot \mu\mathbf{1}) - \tilde{\mathbf{y}}\|_2^2 + \lambda\|(\mathbf{m} \odot \mu\mathbf{1})\|_1, \quad (3)$$

where $\mathbf{1} = [1, 1, \ldots, 1]^T \in \mathbb{R}^N$. In Eq.(3), in contrast to the conventional l_1 optimization of $\boldsymbol{\alpha}$, to solve the problem with QUBO, $\boldsymbol{\alpha}$ is set to be $(\mathbf{m} \odot \mu\mathbf{1})$, where $\mathbf{m}$ is optimized while $\mu\mathbf{1}$ is a fixed vector with all elements set to a hyperparameter $\mu > 0$. This leads to an approximate non-negative l_1 optimization problem with the loss term simplified as

$$\begin{aligned}
\mathcal{L} &= \|\mu\mathbf{D}_l\mathbf{m} - \tilde{\mathbf{y}}\|_2^2 + \lambda\mu\|\mathbf{m}\|_1 \\
&= \mathbf{m}^T \left(\mu^2\mathbf{D}_l^T\mathbf{D}_l\right)\mathbf{m} - 2\left(\mu\mathbf{D}_l^T\tilde{\mathbf{y}}\right)^T\mathbf{m} + \lambda\mu\mathbf{1}^T\mathbf{m} + \mathrm{c} \\
&= \mathbf{m}^T \left(\mu\mathbf{D}_l^T\mathbf{D}_l\right)\mathbf{m} + \left(-2\mathbf{D}_l^T\tilde{\mathbf{y}} + \lambda\mathbf{1}\right)^T\mathbf{m} + \mathrm{c},
\end{aligned} \quad (4)$$

where 'c' symbolizes constant w.r.t optimization variable. Comparing Eq.(4) with the basic form of the QUBO problem in Eq.(1) *i.e.*, $\mathcal{L} = \mathbf{m}^T\mathbf{Q}\mathbf{m} + \mathbf{b}^T\mathbf{m}$, the matrix $\mathbf{Q}$ and the vector $\mathbf{b}$ are given by

$$\begin{aligned}
\mathbf{Q} &= \mu\mathbf{D}_l^T\mathbf{D}_l \\
\mathbf{b} &= -2\mathbf{D}_l^T\tilde{\mathbf{y}} + \lambda\mathbf{1}.
\end{aligned} \quad (5)$$

4.2. Algorithms

Having defined the basic formulation for SISR in the previous section, three algorithms are outlined—including ours, and tested. With all algorithms being based on the same basic formulation, they differ by the optimization algorithm used for sparse approximation. Hence, they are referred to by their optimization algorithm. Concretely, the optimization algorithms used are *(i)* **Lasso Regression**, *(ii)* **Classical Annealing** and *(iii)* **Quantum Annealing**. The details of each algorithm are discussed in the remaining part of this section.

(i) **Lasso Regression.** The overall Lasso Regression algorithm is shown in **Algorithm** 1. With trained dictionaries $\mathbf{D}_h$ and $\mathbf{D}_l$, the overall algorithm takes a LR image Y as input and outputs the SR image X. Sparse coding is carried out in a patch-wise manner with a l_1 optimization problem solved for each patch. To solve the l_1 optimization, the `scikit-learn` Python machine learning library's `linear_model.Lasso` solver is used, featuring a `C`-optimized coordinate descent algorithm. Upon combining patch-wise estimates into a single SR image estimate, the image is subject to a backprojection procedure with 100 iterations for convergence to a stable solution with respect to a global reconstruction constraint $X^* = \operatorname{argmin}_X \|X - X_0\|$ s.t. $(HX) \downarrow = Y$, where H is a blurring filter and $\downarrow$ is a downsampling operation.

Algorithm 1 Lasso Regression

Input: Dictionaries $\mathbf{D}_h$ and $\mathbf{D}_l$, and LR image Y
Extract feature map $\tilde{Y} = F(Y)$
for patch $\tilde{\mathbf{y}}_i$ in $\tilde{Y}$ **do**
    $\boldsymbol{\alpha}^* \leftarrow \operatorname{argmin}_{\boldsymbol{\alpha}} \|\mathbf{D}_l\boldsymbol{\alpha} - \tilde{\mathbf{y}}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1$
    $\mathbf{x}_i \leftarrow \mathbf{D}_h\boldsymbol{\alpha}^*$
    $X_0 \leftarrow \texttt{update}(X_0, \mathbf{x}_i)$
end for
$X^* \leftarrow \operatorname{argmin}_X \|X - X_0\|$ s.t. $(HX)\downarrow = Y$
Output: Super-resolution image X^*

(ii) **Classical Annealing.** As a step towards adapting SISR via sparse representation to be compatible with AQC, an algorithm based on solving QUBO problems is developed. Adapted from **Algorithm 1**, the Classical Annealing algorithm (**Algorithm 2**) uses a QUBO-based formulation with the QUBO problem solved conventionally using CPUs via a simulated annealing algorithm from the `qubovert` library implemented in C. Since in a QUBO problem binary variables are optimized, the binary sparse coding is carried out with the sparse coefficient vector set to be $(\mathbf{m} \odot \mu\mathbf{1})$ as per the formulation in Sec.(4.1). Following Eq.(5), to facilitate binary sparse coding, appropriate constructions are made for the problem matrix $\mathbf{Q}$ and bias $\mathbf{b}$ of the QUBO problem.

Algorithm 2 Classical Annealing

Input: Dictionaries $\mathbf{D}_h$ and $\mathbf{D}_l$, and LR image Y
Extract feature map $\tilde{Y} = F(Y)$
for patch $\tilde{\mathbf{y}}_i$ in $\tilde{Y}$ **do**
    $\mathbf{Q} \leftarrow \mu\mathbf{D}_l^T\mathbf{D}_l$
    $\mathbf{b} \leftarrow -2\mathbf{D}_l^T\tilde{\mathbf{y}}_i + \lambda\mathbf{1}$
    $\mathbf{m}^* \leftarrow \operatorname{argmin}_{\mathbf{m}} \mathbf{m}^T\mathbf{Q}\mathbf{m} + \mathbf{b}^T\mathbf{m}$
    $\boldsymbol{\alpha}^* \leftarrow \mu\mathbf{m}^*$
    $\mathbf{x}_i \leftarrow \mathbf{D}_h\boldsymbol{\alpha}^*$
    $X_0 \leftarrow \texttt{update}(X_0, \mathbf{x}_i)$
end for
$X^* \leftarrow \operatorname{argmin}_X \|X - X_0\|$ s.t. $(HX)\downarrow = Y$
Output: Super-resolution image X^*

(iii) **Quantum Annealing.** The Quantum Annealing algorithm (**Algorithm 3**) represents a way for SISR via sparse representation to be solved using QPUs. It is a modification of the *Qbsolv* algorithm developed by D-Wave [5]. Quantum Annealing uses a `CreateQUBO` function (**Algorithm 4**) to generate a set of QUBO problems to be solved. In **Algorithm 4**, the typical sparse coding problem $\operatorname{argmin}_{\mathbf{m}} \mathbf{m}^T\mathbf{A}\mathbf{m} + \mathbf{b}^T\mathbf{m}$ is generated patch-wise from the feature-extracted LR image $\tilde{Y}$ and the LR dictionary $\mathbf{D}_l$. The sparse coding problem is initially solved using a MST2

multistart tabu search algorithm from the `dwave-tabu` library. With an initial solution $\mathbf{m}^*$ obtained, a heuristic `EnergyImpactDecomposer` selects the variables of $\mathbf{m}$ that are expected to cause the most change in the energy $E(\mathbf{m}) = \mathbf{m}^T\mathbf{A}\mathbf{m} + \mathbf{b}^T\mathbf{m}$. The selected variables, numbering 32, are kept track of by `index`.

Algorithm 3 Quantum Annealing

Input: Dictionaries $\mathbf{D}_h$ and $\mathbf{D}_l$, and LR image Y
$\{Z_Q\}, \{E_Q\}, \{O_Q\} \leftarrow \emptyset$
Extract feature map $\tilde{Y} = F(Y)$
$\{\mathbf{Q}\}, \mathbf{M}_0, \texttt{index} \leftarrow \texttt{CreateQUBO}(\tilde{Y}, \mathbf{D}_l)$
for $\mathbf{Q}$ in $\{\mathbf{Q}\}$ **do**
    $Z_Q, E_Q, O_Q \leftarrow \texttt{SolveQUBO}(\mathbf{Q}, N_{reads})$
    $\{Z_Q\} \leftarrow \{Z_Q\} \cup Z_Q$
    $\{E_Q\} \leftarrow \{E_Q\} \cup E_Q$
    $\{O_Q\} \leftarrow \{O_Q\} \cup O_Q$
end for
for patch $\tilde{\mathbf{y}}_i$ in $\tilde{Y}$ **do**
    $Z_i, E_i, O_i \leftarrow \texttt{extract}(\{Z_Q\}, \{E_Q\}, \{O_Q\})$
    $\mathbf{m}_{0i} \leftarrow \texttt{extract}(\mathbf{M}_0)$
    $\mathbf{x}_i \leftarrow 0$
    $\mathcal{Z}_i \leftarrow 0$
    for read $\mathbf{z}_{ij}$ in Z_i **do**
        $\boldsymbol{\alpha}^* \leftarrow \mu\mathbf{m}_{0i}$
        $\boldsymbol{\alpha}^*[\texttt{index}_i] \leftarrow \mu\mathbf{z}_{ij}$
        $p_{ij} \leftarrow O_{ij}\exp\left(-\beta E_{ij}\right)$
        $\mathbf{x}_i \leftarrow \mathbf{x}_i + p_{ij}\mathbf{D}_h\boldsymbol{\alpha}^*$
        $\mathcal{Z}_i \leftarrow \mathcal{Z}_i + p_{ij}$
    end for
    $\mathbf{x}_i \leftarrow \mathbf{x}_i/\mathcal{Z}_i$
    $X_0 \leftarrow \texttt{update}(X_0, \mathbf{x}_i)$
    $\mathcal{H}_i = -\sum_j p_{ij}\log(p_{ij})$
end for
$X^* \leftarrow \operatorname{argmin}_X \|X - X_0\|$ s.t. $(HX)\downarrow = Y$
Output: Super-resolution image X^*

Subsequently, a *clamping* step is applied whereby the non-selected binary variables of $\mathbf{m}$ have their values fixed from the tabu-searched $\mathbf{m}^*$ and these values are then used as part of the construction of QUBO subproblems. This leads to the optimization of the 32 highest impact binary variables for each patch. As such, each patch is associated with a QUBO subproblem of size 32 and subproblems are added along diagonals of overall $\mathbf{Q}$ matrices of size 512 as per Fig.2. This construction leads to sparse $\mathbf{Q}$ matrices of size 512×512 that are suitable for the sparse QPU topologies.

The set of QUBO problems constructed is then submitted to a D-Wave direct AQC solver, represented by the `SolveQUBO` function. Each QUBO problem submitted comes with N_{reads} solutions returned. These solutions are organized in Z_Q such that only unique solutions are present. E_Q and O_Q contain the energy and occurrence of each

Algorithm 4 `CreateQUBO` for Quantum Annealing

Input: LR image features $\tilde{Y}$ and dictionary $\mathbf{D}_l$
for patch $\tilde{\mathbf{y}}_i$ in $\tilde{Y}$ **do**
    $\mathbf{A} \leftarrow \mu \mathbf{D}_l^T \mathbf{D}_l$
    $\mathbf{b} \leftarrow -2\mathbf{D}_l^T \tilde{\mathbf{y}}_i + \lambda \mathbf{1}$
    $\mathbf{m}^* \leftarrow \arg\min_{\mathbf{m}} \mathbf{m}^T \mathbf{A} \mathbf{m} + \mathbf{b}^T \mathbf{m}$ ▷ (Tabu search)
    $\mathbf{M}_0[iN:(i+1)N] \leftarrow \mathbf{m}^*$
    $\text{index}_i \leftarrow \text{EnergyImpactDecomposer}(\mathbf{m}^*, \mathbf{A}, \mathbf{b})$
    $\mathbf{m}_{tmp} \leftarrow \mathbf{m}^*$
    $\mathbf{m}_{tmp}[\text{index}_i] \leftarrow 0$
    $\mathbf{d} \leftarrow 2\mathbf{A}[\text{index}_i]\mathbf{m}_{tmp}$
    $\mathbf{Q} \leftarrow \mathbf{A} + \text{diag}(\mathbf{d} + \mathbf{b})$
    $\{\mathbf{Q}\} \leftarrow \text{AddSubproblem}(\{\mathbf{Q}\}, \mathbf{Q})$ ▷ (Figure 2)
end for
Output: $\{\mathbf{Q}\}, \mathbf{M}_0, \text{index}$

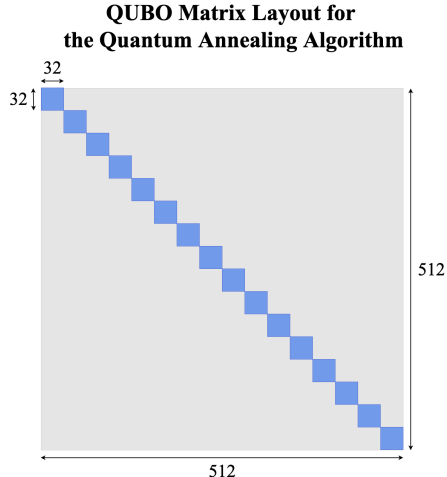**QUBO Matrix Layout for
the Quantum Annealing Algorithm**

Figure 2. QUBO problem matrix layout for Quantum Annealing. The blue parts indicate parts with filled values while the remaining region is not filled with values and as such left out of embedding on QPUs.

unique solution. Together, these returned objects organized in the form of sets $\{Z_Q\}, \{E_Q\}, \{O_Q\}$ provide a probabilistic distribution over optimal solutions.

Having obtained $\{Z_Q\}, \{E_Q\}, \{O_Q\}$; Z_i, E_i, O_i are extracted for each patch i and they are combined with the tabu-searched initial patch sparse coefficient estimate $\mathbf{m}_{0i}$. The HR $\mathbf{x}_i$ is generated by taking the weighted mean over the unique solutions in Z_i. The weight associated with each unique solution is a *probability* determined by the energy and occurrence as $p_{ij} = (O_{ij}/\mathcal{Z}_i)\exp(-\beta E_{ij})$, where $\mathcal{Z}_i = \sum_j O_{ij}\exp(-\beta E_{ij})$. Given the p_{ij} weights, the patch entropy $\mathcal{H}_i = -\sum_j p_{ij}\log(p_{ij})$ can be defined, providing a metric for uncertainty quantification.

5. Experiments

5.1. Training & Evaluation Details

To train the dictionaries $\mathbf{D}_l$ and $\mathbf{D}_h$, the same procedure and dataset as in Yang et al. [39] are used. An essential result from Yang et al. [39] is that dictionaries generated by randomly sampling raw patches from images of similar statistical nature combined with a sparse representation prior are sufficient to generate high quality reconstructions. The training procedure samples raw patches from training images, prunes off patches of small variances and uses an online dictionary learning algorithm [24] to generate $\mathbf{D}_l$ and $\mathbf{D}_h$. In this work, to accommodate the sparse topology of quantum annealers, a relatively small number of atoms N i.e., 128, are learned by $\mathbf{D}_l$ and $\mathbf{D}_h$. The dataset used to train $\mathbf{D}_l$ and $\mathbf{D}_h$ in this work is a dataset consisting of 69 images used by Yang et al. [39]. In addition to training, hyperparameters for our algorithms are optimized. For Lasso Regression, $\lambda = 10^{-5}$ is used. The annealing-based algorithms share a common set of sparse coding hyperparameters with $\lambda = 0.1$ and $\mu = 0.05$. In Quantum Annealing, $N_{reads} = 100$ is used as the number of samples obtained

from AQC solvers.

The test data used for evaluation consist of frequently used data in the image processing domain such as the Lenna image and the Set5 image test set. From these data, 3 test cases are set up. The first test case uses a 45×60 cropped region from the Lenna image as the HR ground-truth, with the corresponding downsampled 15×20 region as the LR image (Fig.1). The second and third test cases are simply the full Lenna image with its downsampled LR version and Set5 respectively. The first test case of the Lenna image region serves as the basis of a run-time analysis comparing the sparse coding algorithms. As part of the evaluations, a DL method is compared against the sparse coding algorithms. For this, we used the SwinIR [22], a current state-of-the-art PSNR-oriented SISR model. The version of SwinIR used is a model pretrained on the DIV2K [2] image set with a training patch size of 48.

Due to the varying nature of the algorithms developed, evaluations are naturally carried out on different hardware. Classical algorithms and client-side data processing are carried out on Intel Core 4.00GHz CPUs with 4 cores and 32GB RAM, while the Quantum Annealing algorithm which accesses QPUs directly uses the D-Wave Advantage 6.1 QPU with 5760 qubits.

The accuracy metric used in this work is the Y-channel PSNR between the SR prediction and the HR ground-truth. All downsampling operations reduce image height and width by a factor of 3 and correspondingly all the SR algorithms developed carry out upsampling by a factor of 3.
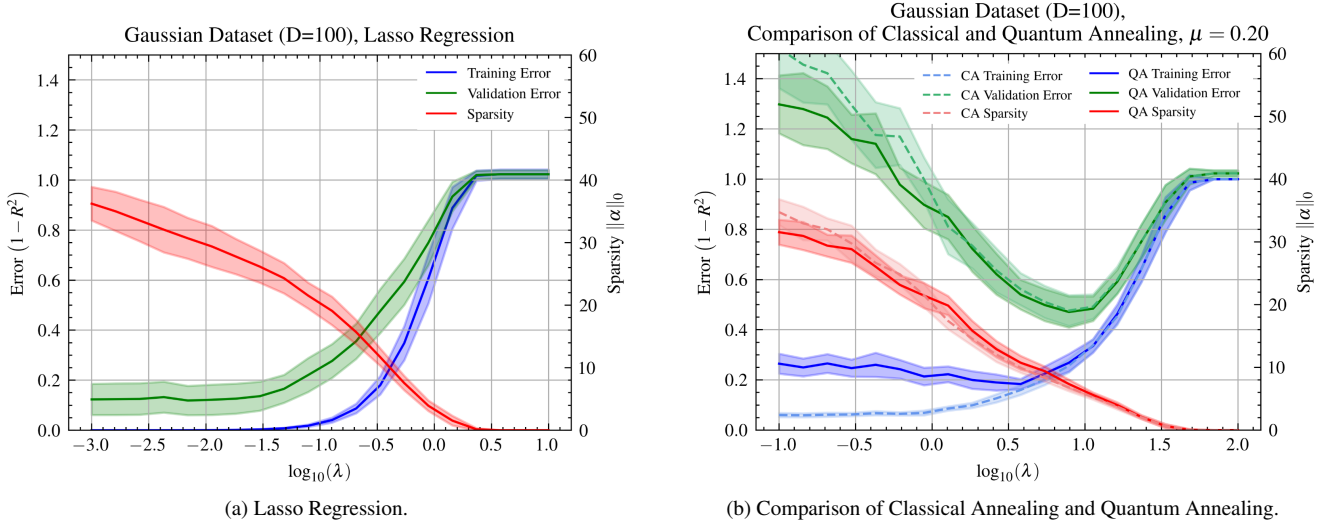
Figure 3. **Left:** (a) Numerical results for sparse coding performed by Lasso Regression on 20 datasets generated from a common random process. **Right:** Numerical results for sparse coding performed by Classical Annealing (dashed lines) and Quantum Annealing (solid lines) on 20 datasets generated from a common random process. The result for Quantum Annealing is very similar to the result for Classical Annealing especially for the sparsity $\|\boldsymbol{\alpha}\|_0 \leq 10$. Note: The shaded regions in the plots show the observed standard deviation.

5.2. Numerical Experiments

Prior to evaluation on real images, the algorithms developed were first tested on numerical datasets to provide insights about their predictive performance. For all numerical experiments detailed in this section, simulated datasets are generated from a common underlying random process. Specifically, each dataset is of the form of a data matrix $\mathbf{X}_{all} \in \mathbb{R}^{756 \times 100}$ with all elements sampled from the standard Gaussian distribution $\mathcal{N}(0, 1)$. Subsequently, a non-negative random vector $\boldsymbol{\alpha}_{small} \in \mathbb{R}^{10}_{\geq 0}$ is sampled from $|\mathcal{N}(0, 1)|$. $\boldsymbol{\alpha}_{small}$ is then multiplied with a $\mathbb{R}^{756 \times 10}$ slice of $\mathbf{X}_{all}$ to yield the target vector $\mathbf{y}_{all} \in \mathbb{R}^{756}$. The overall dataset $(\mathbf{X}_{all}, \mathbf{y}_{all})$ is partitioned into training and validation sets $(\mathbf{X}_{train}, \mathbf{y}_{train})$ and $(\mathbf{X}_{val}, \mathbf{y}_{val})$ with sizes 36 and 720 respectively. The resulting setup simulates the sparse coding of a $\mathbb{R}^{36 \times 100}$ dictionary with an optimal sparsity of 10 atoms, as the algorithms tested seek to isolate the 10 features corresponding to $\boldsymbol{\alpha}_{small}$ from the noisy data.

The numerical results for Lasso Regression and Classical Annealing are shown in Fig.(3a), and the result for Quantum Annealing is shown in Fig.(3b). In Fig.(3a) a range of values for the sparsity constraint λ is evaluated and the prediction error and the $\boldsymbol{\alpha}$ sparsity level corresponding to each λ value are shown. The error metric used for evaluating the predictions $\hat{\mathbf{y}}_{train}$ and $\hat{\mathbf{y}}_{val}$ is $1 - R^2$ where R^2 is the coefficient of determination.

Fig.(3b) shows that Classical Annealing and Quantum Annealing have very similar numerical behavior, no less due to the mathematical similarity of their formulations. For both algorithms, at the known true optimal sparsity level of

$\|\boldsymbol{\alpha}\|_0 = 10$, the validation errors are at around 0.5. At around $\|\boldsymbol{\alpha}\|_0 = 10$, the mean validation error of either annealing-based algorithm lies well within the 1σ region of the other while there is significant overlap between the 1σ regions. Thus, there is strong statistical grounding for the similarity of numerical behavior of the two algorithms. In comparison, while Lasso Regression achieves better prediction results at low sparsities, towards the optimal sparsity $\|\boldsymbol{\alpha}\|_0 = 10$, the validation error converges to those of the annealing-based algorithms. These observations provide evidence that the annealing-based algorithms have similar ability to recover the meaningful features corresponding to $\boldsymbol{\alpha}_{small}$, hence similar sparse coding performance to Lasso Regression, despite the more limited set of representations achievable by binary sparse coding.

5.3. Classical & Quantum Algorithm Comparison

The 3 algorithms in Sec.(4.2) are evaluated and compared against a bicubic interpolation baseline and the SwinIR [22] DL model. The Y-channel PSNRs of these 5 models in total evaluated on the test cases of the Lenna image region, the full Lenna image and the Set5 test set are displayed in Tab.(1).

Tab.(1) shows that for all test cases, Quantum Annealing achieves very similar accuracy as Classical Annealing. In fact, the two algorithms achieve the same PSNR metrics for the larger test cases at a precision of 2 decimal places. This demonstrates that the similar numerical behavior between the two algorithms shown in Fig.(3b) is replicated when applied to images. Another observation from Fig.(3a)

and Fig.(3b) reflected in Tab.(1) is that the predictive performance of annealing-based algorithms at the very least matches that of Lasso Regression. Going beyond Fig.(3a) and Fig.(3b), the results for the larger test cases in Tab.(1) furthermore suggest the possibility of superior performance of annealing-based sparse coding over conventional sparse coding when the data is in the form of real images as opposed to simulated data as in Sec.(5.2). Nevertheless, the sparse coding approaches have accuracies that fall short of the current state-of-the-art SwinIR DL model.

Model	PSNR		
	(a) Lenna reg. (dB)	(b) Lenna (dB)	(c) Set5 (dB)
Bicubic	28.31	30.62	29.35
Lasso Regress.	30.22	31.58	30.44
Classical Ann.	30.20	31.70	30.61
Quantum Ann.	30.19	31.70	30.61
SwinIR (DL)	**31.42**	**33.29**	**33.89**

Table 1. PSNR results of the algorithms evaluated on (a) cropped region of the Lenna image (Fig.4), (b) the full Lenna image Fig.5 and (c) the Set5 image set. The magnification factor is 3. The quantum approach generally shows favorable results compared to the classical Lasso. Moreover, our initial primitive quantum experiments for SISR shows encouraging results compared to the state-of-the-art deep learning (DL) method.
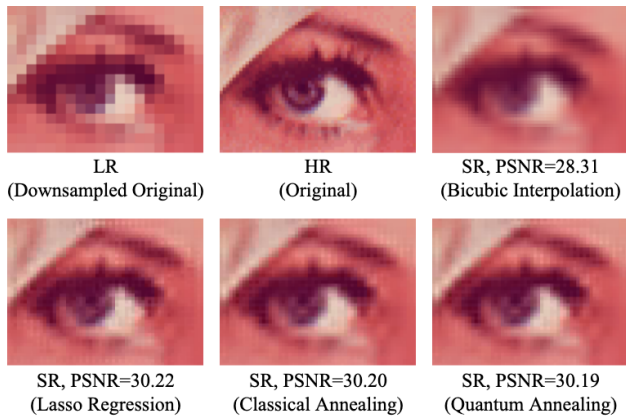

LR
(Downsampled Original)

HR
(Original)

SR, PSNR=28.31
(Bicubic Interpolation)

SR, PSNR=30.22
(Lasso Regression)

SR, PSNR=30.20
(Classical Annealing)

SR, PSNR=30.19
(Quantum Annealing)

Figure 4. Results on a cropped region of the Lenna image with a magnification factor of 3. **Top row.** *From left to right:* LR, HR and SR using Bicubic interpolation. **Bottom row.** *From left to right:* SR using Lasso Regression, Classical Annealing and Quantum Annealing respectively.

The run-times of the algorithms on the other hand differ widely, with the annealing-based algorithms requiring more time to run than Lasso Regression. For purposes of analysis, the total run-time of each algorithm is decomposed into

LR
(Downsampled Original)

SR, PSNR=31.70
(Quantum Annealing)

Figure 5. Result on the full Lenna image using the Quantum Annealing algorithm with a magnification factor of 3. *Left:* LR. *Right:* SR using Quantum Annealing.

5 run-times. They are namely CPU Optimization, Create-QUBO, AQC Preparation, QPU Optimization and Miscellaneous.

The CreateQUBO run-time refers to the time taken to execute the `CreateQUBO` function in Quantum Annealing (**Algorithm** 4). Executions of `CreateQUBO` are done exclusively on client-side CPUs. The CPU Optimization run-time refers specifically to the sum of the time taken for all client-side CPU sparse coding steps. As such, it is applicable to executions of the `scikit-learn linear_model.Lasso` function for Lasso Regression and the `qubovert.sim.anneal_qubo` function for Classical Annealing, and excludes QPU-based optimization. On the other hand, AQC Preparation and QPU Optimization are run-times applicable to Quantum Annealing. When a QUBO problem is submitted via the D-Wave Leap interface, a period of time is required for communication, scheduling and assignment of the problem to a specific QPU. Subsequently, the QUBO problem is programmed onto a QPU, subject to annealing, sampling and then post-processed. In this context, the QPU Optimization run-time refers to the combined amount of time taken for programming, annealing and sampling on a QPU, while the AQC Preparation run-time refers to the combined amount of time taken for tasks of communication, scheduling, assignment, postprocessing and other overheads done by the server-side CPU.

Tab.(2) shows that while Quantum Annealing achieves speed up over Classical Annealing, it is far from matching Lasso Regression's compute time. To improve the speed of Quantum Annealing, one possible approach is to replace the tabu search in **Algorithm** 4 with a random initialization and omit the use of `EnergyImpactDecomposer` by instead applying quantum annealing over all subsets of m. This approach would drastically simplify the `CreateQUBO` function but would evidently require more quantum computing

Model	CPU Opt. (s)	CreateQUBO (s)	AQC Prep. (s)	QPU Opt. (s)	Misc. (s)	Total Run-time (s)
Lasso Regression	4.80	-	-	-	0.09	4.89
Classical Annealing	658.24	-	-	-	0.12	658.36
Quantum Annealing	-	31.68	55.42	1.58	0.92	89.60

Table 2. Run-time results of the algorithms evaluated on a cropped region of the Lenna image (Fig.4). The total run-time is broken down into 5 constituent run-times, namely CPU Optimization, CreateQUBO, AQC Preparation, QPU Optimization and Miscellaneous. Note that some run-times are not applicable to each algorithm.

resources in terms of access time. The AQC Preparation run-time on the other hand, could not be reduced without fundamental changes to the client-server framework on which this work is based. Nevertheless, should local QPU access be possible, the client-server communication overhead could in principle be eliminated for a much smaller total run-time in which the QPU Optimization run-time becomes the most significant constituent.

5.4. Robust Prediction & Uncertainty Estimation

As discussed in Sec.(4.2), the D-Wave AQC solver used in Quantum Annealing provides a means to carry out repeated sampling of a QUBO problem to obtain a distribution of solutions, also known as *reads*. This property provides Quantum Annealing with the advantage of providing a rapid means for robust prediction and uncertainty estimation compared to classical algorithms.
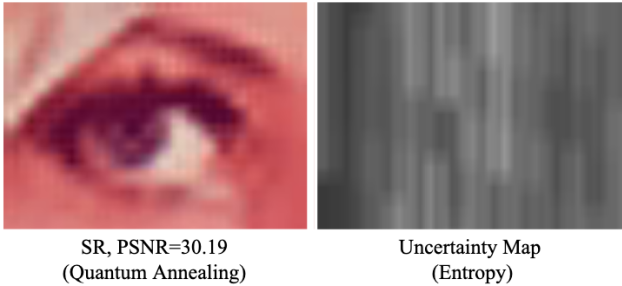

SR, PSNR=30.19
(Quantum Annealing)

Uncertainty Map
(Entropy)

Figure 6. *Left:* Result on a cropped region of the Lenna image (Fig.4) using the Quantum Annealing algorithm. *Right:* Uncertainty map of the SR image prediction using entropy as the metric for uncertainty. The entropy is computed for each patch i as $\mathcal{H}_i = -\sum_j p_{ij} \log(p_{ij})$ where p_{ij} is the probability associated with a particular annealing read sample $\mathbf{z}_{ij}$.

Taking the expectation over a distribution of predictions can give a robust prediction, as done in methods such as Bayesian neural networks [23]. β effectively determines the number of reads used to generate the final estimate. For example in the limit $\beta \to \infty$, the Quantum Annealing algorithm merely returns a point estimate *i.e.*, the lowest energy solution. On the other hand, when β is smaller, more equal weighting is placed on solutions. At a certain optimal β, the balance between uniqueness and diversity can yield optimally robust estimates. However, due to limited quantum computing resources, insufficient evidence is observed for the advantage offered by such robust prediction from the limited evaluations. As such, further relevant numerical experiments and more extensive image evaluations may be the subject of future work.

Fig.(6) shows a visualization of the result of uncertainty quantification using Quantum Annealing, where the uncertainty is represented by a map of the entropy. The vertically oriented stripes in Fig.(6) are due to the column-major nature of the patch-wise iteration in **Algorithm** 3, where neighboring patches along a column are processed in the same QUBO problem.

6. Conclusion

Inspired by the early success of sparse coding in computer vision, in this paper, a quantum algorithm for the SISR problem is proposed, discussed, and shown to have a close relationship with the quadratic unconstrained assignment problem—a popular quantum optimization formulation. The summarized quantum annealing algorithm for SISR provides encouraging initial results. While the proposed algorithm is distant from the deep learning-based state-of-the-art image super-resolution result, as an initial demonstration of carrying out SISR with quantum hardware, this work has provided some evidence for superior sparse coding performance of annealing-based methods over conventional sparse coding and insights on how to reduce run-times of the former to comparable levels. Furthermore, the algorithm design, including aspects such as sparse binary coding, QUBO problem layout structuring, and multi-read sampling, can also be applied to the proposed problem settings with advantages offered by quantum annealing, such as superior sparse coding performance, faster robust prediction, and uncertainty estimation.

8

References

[1] Adetokunbo Adedoyin, John Ambrosiano, Petr Anisimov, Andreas Bärtschi, William Casper, Gopinath Chennupati, Carleton Coffrin, Hristo Djidjev, David Gunter, Satish Karra, et al. Quantum algorithm implementations for beginners. *arXiv preprint arXiv:1804.03719*, 2018. 2

[2] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 5

[3] Syed Muhammad Arsalan Bashir, Yi Wang, Mahrukh Khan, and Yilong Niu. A comprehensive review of deep learning-based single image super-resolution. *PeerJ Computer Science*, 7:e621, 2021. 2

[4] Marcel Seelbach Benkner, Zorah Lähner, Vladislav Golyanik, Christof Wunderlich, Christian Theobalt, and Michael Moeller. Q-match: Iterative shape matching via quantum annealing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7586–7596, 2021. 2

[5] M Booth, SP Reinhardt, and A Roy. Partitioning optimization problems for hybrid classical/quantum execution. Technical report, Technical Report, http://www. dwavesys. com, 2017. 4

[6] Honggang Chen, Xiaohai He, Linbo Qing, Yuanyuan Wu, Chao Ren, Ray E Sheriff, and Ce Zhu. Real-world single image super-resolution: A brief review. *Information Fusion*, 79:124–145, 2022. 2

[7] Shengyang Dai, Mei Han, Wei Xu, Ying Wu, and Yihong Gong. Soft edge smoothness prior for alpha channel super resolution. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2

[8] Anh-Dzung Doan, Michele Sasdelli, David Suter, and Tat-Jun Chin. A hybrid quantum-classical algorithm for robust fitting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 417–427, 2022. 2

[9] Anh-Dzung Doan, Michele Sasdelli, David Suter, and Tat-Jun Chin. A hybrid quantum-classical algorithm for robust fitting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 417–427, June 2022. 2

[10] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016. 2

[11] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. 2004. 3

[12] Sina Farsiu, M Dirk Robinson, Michael Elad, and Peyman Milanfar. Fast and robust multiframe super resolution. *IEEE transactions on image processing*, 13(10):1327–1344, 2004. 2

[13] Dario Fuoli, Zhiwu Huang, Shuhang Gu, Radu Timofte, Arnau Raventos, Aryan Esfandiari, Salah Karout, Xuan Xu, Xin Li, Xin Xiong, et al. Aim 2020 challenge on video extreme super-resolution: Methods and results. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 57–81. Springer, 2020. 1

[14] Vladislav Golyanik and Christian Theobalt. A quantum computational approach to correspondence problems on point sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9182–9191, 2020. 2

[15] Russell C Hardie, Kenneth J Barnard, and Ernest E Armstrong. Joint map registration and high-resolution image estimation using a sequence of undersampled images. *IEEE transactions on Image Processing*, 6(12):1621–1633, 1997. 2

[16] Hsieh Hou and H Andrews. Cubic splines for image interpolation and digital filtering. *IEEE Transactions on acoustics, speech, and signal processing*, 26(6):508–517, 1978. 2

[17] Shuxian Jiang, Keith A Britt, Alexander J McCaskey, Travis S Humble, and Sabre Kais. Quantum annealing for prime factorization. *Scientific reports*, 8(1):1–9, 2018. 2

[18] Zhang Jiang, Eleanor G Rieffel, and Zhihui Wang. Near-optimal quantum circuit for grover's unstructured search using a transverse field. *Physical Review A*, 95(6):062317, 2017. 2

[19] S Kathiravan and J Kanakaraj. An overview of sr techniques applied to images, videos and magnetic resonance images. *Smart CR*, 4(3):181–201, 2014. 2

[20] Yongwoo Kim, Jae-Seok Choi, and Munchurl Kim. A real-time convolutional neural network for super-resolution on fpga with applications to 4k uhd 60 fps video services. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8):2521–2534, 2018. 2

[21] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017. 1

[22] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 2, 5, 6

[23] David JC MacKay. A practical bayesian framework for back-propagation networks. *Neural computation*, 4(3):448–472, 1992. 8

[24] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1), 2010. 5

[25] Catherine C McGeoch, Richard Harris, Steven P Reinhardt, and Paul I Bunyk. Practical annealing-based quantum computing. *Computer*, 52(6):38–46, 2019. 2

[26] Catherine C McGeoch and Cong Wang. Experimental evaluation of an adiabatic quantum system for combinatorial optimization. In *Proceedings of the ACM International Conference on Computing Frontiers*, pages 1–11, 2013. 2

[27] Natacha Kuete Meli, Florian Mannel, and Jan Lellmann. An iterative quantum approach for transformation estimation

from point sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 529–537, June 2022. 2

[28] Florian Neukart, Gabriele Compostella, Christian Seidel, David Von Dollen, Sheir Yarkoni, and Bob Parney. Traffic flow optimization using a quantum annealer. *Frontiers in ICT*, 4:29, 2017. 2

[29] EC Pasztor, Owen T Carmichael, and WT Freeman. Learning low-level vision. *International Journal of Computer Vision*, 1(40):25–47, 2000. 2

[30] Eleanor Rieffel and Wolfgang Polak. An introduction to quantum computing for non-physicists. *ACM Computing Surveys (CSUR)*, 32(3):300–335, 2000. 2

[31] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994. 2

[32] Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997. 2

[33] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008. 2

[34] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Computer Vision–ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part IV 12*, pages 111–126. Springer, 2015. 2

[35] Lisa Tse, Peter Mountney, Paul Klein, and Simone Severini. Graph cut segmentation methods revisited with a quantum algorithm. *arXiv preprint arXiv:1812.03050*, 2018. 2

[36] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 2

[37] Yan Wu, Zhiwu Huang, Suryansh Kumar, Rhea Sanjay Sukthanker, Radu Timofte, and Luc Van Gool. Trilevel neural architecture search for efficient single image super-resolution. *arXiv preprint arXiv:2101.06658*, 2021. 1

[38] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5791–5800, 2020. 1

[39] Jianchao Yang, John Wright, Thomas Huang, and Yi Ma. Image super-resolution as sparse representation of raw image patches. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 2, 3, 5

[40] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, Nov. 2010. 2

[41] Ren Yang, Radu Timofte, Meisong Zheng, Qunliang Xing, Minglang Qiao, Mai Xu, Lai Jiang, Huaida Liu, Ying Chen, Youcheng Ben, et al. Ntire 2022 challenge on super-resolution and quality enhancement of compressed video: Dataset, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1221–1238, 2022. 1

[42] Yuan-Fu Yang and Min Sun. Semiconductor defect detection by hybrid classical-quantum deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2323–2332, June 2022. 2

[43] Jan-Nico Zaech, Alexander Liniger, Martin Danelljan, Dengxin Dai, and Luc Van Gool. Adiabatic quantum computing for multi object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8811–8822, 2022. 2