An intelligent system architecture based on a dedicated circuit for nearest neighbor searching is developed in this thesis. The nearest neighbor searching is defined as following: given a set \( X \) of \( n \) data points in a real \( d \)-dimension space \( \mathbb{R}^d \), for any query point \( q \in X \), find the closest point in \( X \) to \( q \) according to a distance measurement. The distances are measured using any distance metric such as the Euclidean distance and the Manhattan distance. Nearest neighbor searching plays an importance role in a variety of applications including data mining, pattern recognition, machine learning, data compression, and statistics.

The nearest neighbor searching method is one of the simplest supervised learning algorithms. In supervised learning, a training sample is an ordered pair \(<x, y>\) where \( x (x \in X) \) is a \( d \)-dimension reference and \( y \) is a label; a test sample is a \( d \)-dimension reference with unknown label. In the training stage of the nearest neighbor searching method, every training sample is simply stored in the reference memory. For classifying a test sample, the distance to every training sample is computed; the label paired with a training sample which has the minimum distance is the classification. This method yields an obvious disadvantage that the time complexity of making classifications and the requirement of the storage for the training samples are very high.

Algorithms to improve the nearest neighbor classification have two categories:

1) Speed up of the query time for nearest-neighbor search.

2) Storage-reducing algorithm for training samples.

Several space-partitioning methods (e.g. the KD-tree) have been proposed for accelerating the search speed of the nearest neighbor searching. The approximate nearest neighbor (ANN) searching, which retrieves an approximation as the nearest neighbor, has been successfully applied in web image and 3D object indexing. The parallelization for the nearest neighbor searching problem by GPU
and hardware implementation can significantly improve the searching speed. On the other hand, the storage-reducing algorithm can minimize the computational efforts in the classification stage by reducing the number of references. Several different algorithms, such as the decision trees, the connectionist networks, and the clustering, have been used to produce the specific references in the training stage. These approaches extend the nearest neighbor classification, which has a simple training process.

In this thesis a short/long-term memory mechanism and a clustering-based prototype learning methodology to reduce the storage of references based on a FPGA-implemented coprocessor for nearest Euclidean distance searching are investigated.

In the mechanism of the short/long-term memory, the references are separated into two parts according to a rank criterion which carries out the transition between the short-term and long-term memory. The references are temporarily stored in the short-term part while they can be memorized for a longer time in the long-term part without receiving the direct influence from the input samples. The rank is the index of a memory which contains the address of reference. The reference is moved to a relatively stable area when a new input sample is matched with the current reference. The occurrence of matching is the nearest neighbor searching with a FPGA-implemented coprocessor. The developed short/long-term memory mechanism can reduce the storage requirements through the forgetting process in the short-term part. Due to the simplicity and the hardware implementation for nearest neighbor searching, the time complexity is not high in the training stage.

To further reduce the storage of references, a clustering algorithm (i.e. K-means) can sufficiently compress the training data to a much lower order. The specific references, named prototypes, are derived from the average of training samples which are in the same cluster. The K-means algorithm is to partition a set of data samples into $k$ clusters by considering the boundary information of confusing classes. Firstly, randomly given $k$ cluster centroids of real $d$-dimensional space are used as the initial prototypes. Then, every training sample is assigned to its nearest cluster centroid according to the distances among all training samples and the $k$ cluster centroids. Each cluster centroid is updated by computing the average of training samples which are assigned to this cluster. As a consequence, the training samples are replaced by the cluster centroids, so that the storage space in the classification algorithm of test samples is substantially reduced.

The storage-reducing algorithm in this work, the short/long-term memory mechanism and the clustering-based prototype learning methodology, also
decrease the computational cost in the classification stage since the number of
references are reduced by a large factor. All of these discussed approaches require
a fast nearest-distance searching which is a common requirement in both the
training (storage-reducing) and the classification stage. An FPGA-implemented
dedicated circuit is developed for the nearest Euclidean distance searching to
solve the common most time-consuming part of these major learning and
recognition algorithms. A large number of nearest neighbor queries arises when
the K-means algorithm converges to a stable status and when a prediction is
made by nearest-neighbor classifier. Consequently, the main computational cost is
overcome by means of the developed hardware acceleration with parallelizing and
pipelining.

Due to the fact that either the number of input and output layers, the size of the
hidden layer, or the weight values vary from case to case, the hardware neural
networks have very limited usefulness in real-world applications. Therefore, a
flexible intelligent system based on hardware/software (HW/SW) co-design is
developed in this thesis to cope with multiple applications such as machine
learning, pattern recognition, and image segmentation on the same hardware
framework. The flexibility of the developed system comes not only from the
software but also from the hardware since the dimensionality of samples and the
number of clusters are scalable so as to further improve the searching speed when
the implemented application has a low dimensional feature vector. The
developed flexible intelligent system has already been benchmarked against the
well known applications of handwritten digit recognition, face recognition and
image segmentation to demonstrate its excellent performance, high flexibility, fast
training speed for storage reduction, short recognition time, good recognition rate
and versatile functionality.