CONFORMAL PREDICTION BASED ACTIVE LEARNING

by

Sergio Matiz Romero

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical and Computer Engineering

Spring 2019

CONFORMAL PREDICTION BASED ACTIVE LEARNING

by

Sergio Matiz Romero

Approved: _____

Kenneth E. Barner, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____

Levi T. Thompson, Ph.D.
Dean of the College of Engineering

Approved: _____

Douglas J. Doren, Ph.D.
Interim Vice Provost for Graduate and Professional Education

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Kenneth E. Barner, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Gonzalo R. Arce, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Javier Garcia-Frias, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Chandra Kambhamettu, Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

I want to express my sincere gratitude to my advisor, Dr. Kenneth E. Barner, for his continuous guidance and academic support during these years. His expertise was invaluable and had a positive impact on the quality of this work. I would also like to thank the members of my dissertation committee: Dr. Gonzalo R. Arce, Dr. Javier Garcia-Frias, and Dr. Chandra Kambhamettu for all their teachings throughout the PhD program. My appreciation extends to my lab-mates for the informal discussions that contributed to my research, and for the many hours spent together in the laboratory. In addition, I would like to thank my family for their unconditional support and advice, which has given me the strength to endure countless difficulties during this journey. Last but not least, I want to thank Laura Arévalo, for her caring love and support during this endeavor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Conformal prediction uses the degree of strangeness (nonconformity) of new data instances to determine the confidence values of new predictions. Conformal predictors are implemented in conjunction with traditional pattern classification algorithms yielding a set of predicted class labels with guaranteed error rate, a property referred to as validity. Different from Bayesian methods, which require prior knowledge of the distribution that generates the data, conformal prediction is only based on the assumption that the data are independent and identically distributed.

Conformal prediction has been shown to improve the performance of pattern classification algorithms, including support vector machines and neural networks, through active learning. Instances are selected based on their level of uncertainty, instead of being selected at random from an unlabeled pool. Moreover, the quality of the confidence values produced by conformal prediction has been demonstrated in the literature through experimentation, verifying the validity property.

Despite these advances, previous work on conformal prediction considers only uncertainty as the selection criterion for active learning. Selecting a batch of $m > 1$ instances based only on uncertainty may result in the selection of similar instances that do not provide additional information. Moreover, outlier detection is crucial to avoid the selection of instances that are not representative of the data.

In light of the above, we propose novel active learning approaches, within the conformal prediction framework, considering uncertainty, diversity, and representativeness, as the selection criteria. Diversity is used to avoid the selection of similar instances, whereas representativeness is used for outlier detection. This work focuses on the application of conformal prediction to image classification. Experiments conducted

on face, object, and emotion recognition databases demonstrate that the proposed active learning approaches improve the performance of a variety of pattern classification algorithms while producing reliable confidence values.

Chapter 1

INTRODUCTION

Conformal prediction (CP) was proposed by Vovk, Shafer and Gammerman [1] based on the principles of algorithmic randomness and transductive inference. CP uses the degree of strangeness (nonconformity) of new data instances to determine the confidence values of new predictions, which indicate the likelihood of a prediction being correct. Predictions accompanied by confidence values are desirable, since they provide information on the reliability of such predictions. Conformal predictors can be implemented on top of traditional pattern classification algorithms, which are referred to as underlying algorithms.

The CP framework yields a set of predicted class labels with guaranteed error rate, a property referred to as *validity*, *i.e.*, the probability that the correct class label is excluded is less than a specified threshold. The applications of conformal prediction include: breast cancer diagnosis, clinical diagnosis and prognosis of depression, arrhythmia detection, and robust face recognition [2, 3, 4, 5].

Different methods have been proposed to obtain confidence values. The theory of Probably Approximately Correct (PAC) learning provides bounds on the predictive error [6]. However, PAC learning has two major drawbacks: first, the data must be clean in order to avoid loose bounds [7], and, second, the bounds obtained through PAC apply to the overall error rate rather than individual test instances.

The Bayesian framework can also be used to obtain confidence values. Bayesian methods are optimal assuming correct knowledge on the generating prior. However, for real world data, prior knowledge on the generating distribution is often not available. Therefore, Bayesian approaches can lead to incorrect confidence values when their

assumptions are violated. This is experimentally demonstrated by Melluish *et al.* in [8].

The CP framework, on the other hand, yields confidence values associated with individual test instances, which is advantageous for online learning strategies. Moreover, unlike Bayesian methods, CP is only based on the assumption that the data are independent and identically distributed. Therefore, no knowledge on the prior is required. The disadvantage of traditional CP is the computational inefficiency of transductive inference, which restricts its applicability.

Transductive conformal prediction for active learning has been reported in the literature. Shen-Shyang Ho *et al.* [9] proposed the query by transduction, which sequentially selects the most informative instances (uncertainty sampling) from the unlabeled pool. Balasubramanian *et al.* [10] proposed a modified version of this technique, which is known as the generalized query by transduction. The novelty of their approach lies in the information indicator, which is based on eigendecomposition. The aforementioned approaches have been shown to enhance the performance of incremental support vector machines (SVM) with applications to image classification.

Although transductive inference is a promising active learning technique, its major drawback is high computational complexity, since the underlying algorithm must be trained every time a new instance is processed. This becomes computationally prohibitive for any approach that requires significantly long training times.

Inductive conformal prediction emerged as an alternative to transductive inference [1]. This approach only requires that the underlying algorithm be employed once to generate a classification rule, which is then used for active or online learning. Furthermore, inductive conformal prediction satisfies the validity property. The application of inductive conformal predictors (ICP) to decision trees is studied in [11]. Papadopoulos *et al.* [12] applied ICP to neural networks, verifying the validity property through experimentation. Moreover, the authors perform active learning based on uncertainty (informativeness) to improve the performance of neural networks, with applications to image classification. ICP for ensembles of neural networks is studied

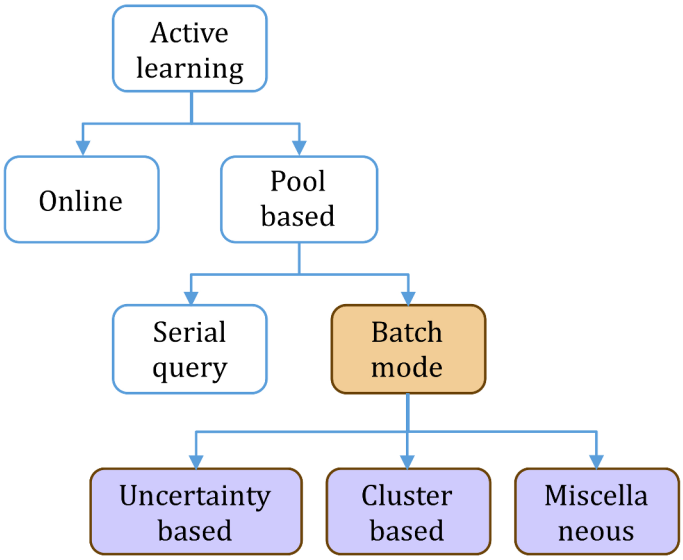


Figure 1.1: Active learning categories.

in [13]. Although the efforts mentioned above are shown to enhance performance, those techniques consider only uncertainty as the selection criterion for active learning, which is only optimal for the selection of one instance at each iteration. Selecting a batch of $m > 1$ instances based solely on uncertainty may result in the selection of similar instances that do not provide additional information.

Active learning algorithms automatically select appropriate data instances to train a classifier reducing the cost (human effort) associated with annotation. Active learning has been extensively applied in domains like classification, image segmentation and information retrieval [14, 15, 16, 17]. Active learning can be roughly divided into two categories: online and pool based, as shown in Fig. 1.1. In online active learning, the learner processes data instances sequentially, as they are observed, and the model has to decide whether or not to query the observed instance to update the hypothesis. Pool based active learning is further divided into serial query based active learning and batch mode active learning. In a serial query based active learning system, the classifier is updated after every single query [18, 19, 20]. This approach is time consuming since the model needs to be retrained frequently. Batch mode active learning techniques

address this issue by selecting multiple instances at a time from the unlabeled pool for annotation [14, 21, 22]. This work focuses on batch mode active learning with applications to image classification.

Batch mode active learning based on both uncertainty and diversity has been shown to improve the performance of pattern classification algorithms [14, 16, 22, 23, 24, 25, 26, 27], avoiding the selection of similar instances that do not provide additional information. Several approaches based on similarity measures have been proposed to measure diversity [14, 28]. For instance, K. Brinker [14] proposes a diversity criterion based on the cosine angle distance between two different instances. Z. Y. Gu *et al.* [28] employ the Gaussian kernel to measure the similarity between two instances. Xu *et al.* [21] apply clustering to measure diversity. Shi *et al.* [16] combine spatial coherence with clustering to improve the performance of remote sensing image classification. Chakraborty *et al.* [26] combine entropy with diversity in a single query function, solving the active leaning problem using quadratic optimization. However, query functions based only on uncertainty and diversity may lead to the selection of outliers that are not representative of the data. Uncertainty and information density (representativeness) have been combined in a single query function to select instances that are both informative and representative [17, 29, 30, 31, 32, 33]. Li *et al.* [30] propose a systematic way for measuring and combining uncertainty and representativeness of unlabeled instances for active learning. Wang *et al.* [32] combine clustering with active/semi-supervised learning to select instances that are representative and discriminative. Du *et al.* [33] derive a robust multi-label active learning algorithm based on the maximum correntropy criterion, merging uncertainty and representativeness in a single optimization problem.

Figure 1.2 divides batch mode active learning mode into work considering uncertainty, diversity, and representativeness. Several authors are placed in the areas where they have made contributions. The question mark in the region where the three criteria overlap indicates that there is little work considering uncertainty, diversity, and representativeness jointly. Recent work by Wang *et al.* [34] combine the three

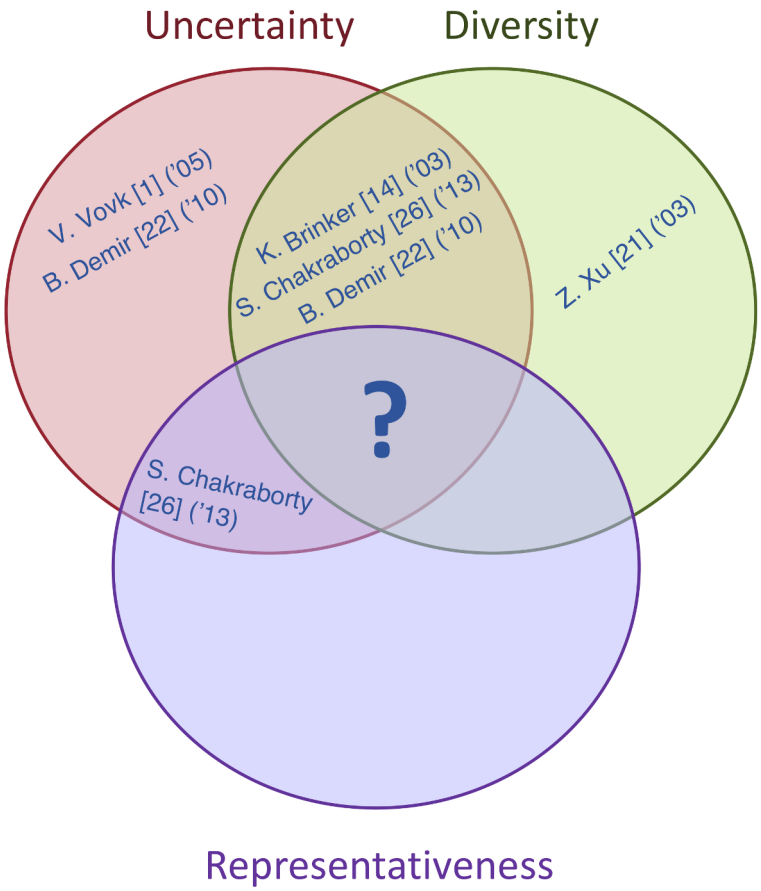


Figure 1.2: Related work on batch mode active learning.

aforementioned criteria via sparse modeling for active learning, however, they consider only SVMs. Kee *et al.* [35] use a weighted sum of three terms, associated with uncertainty, diversity, and representativeness, as a query function for active learning, using random forest as the classifier. Distance metric learning (DML) has recently gained interest in a variety of applications including clustering, classification, and information retrieval [36, 37, 38, 39, 40]. DML produces similarity measures (transformations) that minimize the difference (*e.g.* distance/correlation) of within-class instances and maximize the difference of between-class instances. The performance of k-means clustering is enhanced through DML in [36, 40]. Weinberger *et al.* [37] propose a large-margin

DML algorithm based on the Mahalanobis distance to improve the performance of k-nearest neighbor classification.

Machine learning algorithms, such as support vector machines (SVMs), sparse coding, and convolutional neural networks (CNNs), have recently gained interest in a variety of problems in image processing and computer vision, including face recognition, classification, and image denoising [41, 42, 43, 44, 45, 46, 47, 48, 49, 50]. Support vector machines have received ample treatment being both theoretically well founded and showing excellent generalization performance in practice [41, 42]. Sparse coding algorithms incorporating class label information in the objective function have been show to produce state-of-the-art results for image classification [43, 51]. Moreover, CNNs have led to a series of breakthroughs in image classification. LeCun *et al.* [46] developed a multilayer CNN, referred to as LeNet-5, for classification of handwritten digits. Krizhevsky *et al.* [52] propose a classic CNN architecture, referred to as AlexNet, showing significant improvements upon previous methods for image classification.

Despite these advances, traditional pattern classification algorithms produce simple predictions, without any associated confidence values. Therefore, they require modifications, or additional techniques to be implemented in conjunction with them [53, 54, 55] to perform active learning, since confidence values and a measure of uncertainty are required for that purpose. Moreover, references to active learning considering uncertainty, diversity, and representativeness jointly as the selection criteria are limited in the literature, that is, the majority of existing active learning techniques are prone to either selecting outliers, when representativeness is left out, or similar (redundant) instances, when diversity is not considered. Last but not least, as uncertainty measures differ from each other across different types of classifiers, it becomes difficult to implement the same active learning technique over different pattern classification algorithms without performing modifications.

In light of the above, we propose novel active learning approaches, within the conformal prediction framework, considering uncertainty, diversity, and representativeness, as the selection criteria. Diversity is uased to avoid the selection of similar

instances, whereas representativeness is used for outlier detection. We improve upon previous work on active learning, including that of K. Brinker [14], B. Demir *et al.* [22], and Chakraborty *et al.* [26]. This work focuses on the application of conformal prediction to image classification. By using the CP framework, the proposed techniques offer two advantages: 1) they are flexible across different pattern classification algorithms, since CP produces uncertainty measures that are normalized, regardless of the type classifier being used, 2) in addition to performance enhancement, the proposed approaches produce reliable confidence values.

This work is organized as follows. First, an introduction to conformal prediction, active learning, and the considered databases is presented in Chapter 2. Conformal prediction based active learning for sparse coding classifiers is described in Chapter 3. An active learning algorithm for convolutional neural networks is presented in Chapter 4. Conformal prediction based active learning by linear regression and by nonlinear constrained optimization is described in Chapter 5 and Chapter 6, respectively.

Chapter 2

BACKGROUND

2.1 Conformal prediction

CP uses the nonconformity of new data instances to determine the confidence values of new predictions. For an arbitrary significance level $\epsilon \in [0, 1]$, CP yields a set Ψ^ϵ containing the correct class label of a given data instance with probability $(1 - \epsilon)$, a property referred to as validity [56]. Define a bag of size $n \in \mathbb{R}$ as a collection of n elements, some of which may be identical with each other. Let that bag be denoted as $\wr z_1, \ldots, z_n \int$. Define $z_i = (\mathbf{x}_i, h_i)$, where $\mathbf{x}_i$ represents a data instance and h_i its corresponding class label.

A nonconformity measure $A(\wr z_1, \ldots, z_n \int, z)$ is a function producing a nonconformity score $\alpha \in \mathbb{R}$, representing how different z is from the elements in the bag $\wr z_1, \ldots, z_n \int$. The nonconformity score of an element z_i in $\wr z_1, \ldots, z_n \int$ is obtained as $\alpha_i = A(\wr z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n \int, z_i)$.

In addition, we can measure the conformity of $\mathbf{x}_{n+j}$ to class q using *p-values*, which are defined as [1]:

$$p(\alpha_{n+j}^{(\mathcal{H}_q)}) = \frac{count\{i : \alpha_i > \alpha_{n+j}^{(\mathcal{H}_q)}\}}{n+1}, \tag{2.1}$$

where $\alpha_{n+j}^{(\mathcal{H}_q)}$ is the nonconformity score of $\mathbf{x}_{n+j}$, under the null hypothesis $\mathcal{H}_q$, and $p(\alpha_{n+j}^{(\mathcal{H}_q)})$ is its p-value. Notice that the p-value is highest when all previous nonconformity scores, $\alpha_1, \ldots \alpha_n$, are higher than that of the new instance, $\alpha_{n+j}^{(\mathcal{H}_q)}$, meaning that $\mathbf{x}_{n+j}$ best conforms to class q. CP uses Equation (2.1) to predict the label for $\mathbf{x}_{n+j}$ using the highest p-value. In addition, for each new instance $\mathbf{x}_{n+j}$ and significance level

$\epsilon \in [0,1]$, we form a set of labels $\Psi^{\epsilon}_{n+j} = \{i : p(\alpha^{(\mathcal{H}_i)}_{n+j}) > \epsilon\}$ containing the correct class label for $\mathbf{x}_{n+j}$ with probability $(1-\epsilon)$, according to the validity property.

The p-values are also used to compute the quality of information [18, 10] . Ho and Wechsler [18] define the quality of information (confidence) of instance $\mathbf{x}_{n+j}$ as

$$s(\mathbf{x}_{n+j}) = p^{(1)}_{n+j} - p^{(2)}_{n+j}, \tag{2.2}$$

where $p^{(1)}_{n+j}$ and $p^{(2)}_{n+j}$ are the largest and second largest p-values for instance $\mathbf{x}_{n+j}$, respectively. The uncertainty of an instance $\mathbf{x}_{n+j}$, within the CP framework, can be defined as:

$$I(\mathbf{x}_{n+j}) = 1 - s(\mathbf{x}_{n+j}). \tag{2.3}$$

Conformal predictors can be divided in two types: transductive and inductive. A detailed description of these two approaches is provided below.

2.1.0.1 Transductive Conformal Predictors

Transductive predictors use the actual training set together with a new instance to make a prediction. Let $\mathbf{x}_{n+j}$ be the new instance, $j \in \{1, 2, \dots\}$, M the number of classes, and n the size of the training set. The steps followed by transductive conformal predictors are described below.

- Apply the underlying algorithm to each one of the possible completions

$$(\mathbf{x}_1, h_1), \dots, (\mathbf{x}_n, h_n), (\mathbf{x}_{n+j}, \mathcal{H}_i) \text{ , for } i = 1, \dots, M.$$

- For every null hypothesis $\mathcal{H}_i$, assign a nonconformity score to the training instances $(\mathbf{x}_1, h_1), \dots, (\mathbf{x}_n, h_n)$ and to the pair $(\mathbf{x}_{n+j}, \mathcal{H}_i)$. This results in the sequences

$$\alpha^{(\mathcal{H}_i)}_1, \dots, \alpha^{(\mathcal{H}_i)}_n, \alpha^{(\mathcal{H}_i)}_{n+j} \text{ , for } i = 1, \dots, M.$$

- Compute the p-value for $\mathbf{x}_{n+j}$ based on all possible null hypotheses $\mathcal{H}_i$ by applying (2.1) to the sequences $\alpha^{(\mathcal{H}_i)}_1, \dots, \alpha^{(\mathcal{H}_i)}_n, \alpha^{(\mathcal{H}_i)}_{n+j}$.

- Predict the classification with the largest p-value and calculate the quality of information $I(\mathbf{x}_{n+j})$.

Notice that this approach applies the underlying algorithm M times every time a new instance is processed, which is computationally intensive.

2.1.0.2 Inductive Conformal Predictors

Inductive predictors first learn a classification rule, which is then used to make new predictions. Therefore, the underlying algorithm is applied only once, saving significant computation time. For a new instance $\mathbf{x}_{n+j}$, ICPs perform the following steps:

- Split the training set of size n into two smaller sets, the proper training set of size $\ell = n - r$ and the calibration set of size r, where r is a parameter of the algorithm.
- Employ the proper training set $(z_1, \ldots, z_\ell)$ to generate a classification rule $D_{(z_1,\ldots,z_\ell)}$ using the underlying algorithm.
- Assign a nonconformity score to each one of the instances in the calibration set (using the correct label for each instance). This results in the sequence

$$\alpha_{\ell+1}, \ldots, \alpha_{\ell+r}.$$

- Compute the p-values for $\mathbf{x}_{n+j}$ for all possible null hypotheses $\mathcal{H}_i$ by applying (2.1) to the sequences

$$\alpha_\ell, \ldots, \alpha_{\ell+r}, \alpha_{n+j}^{(\mathcal{H}_i)} \text{ , for } i = 1, \ldots, M.$$

- Predict the classification with the largest p-value and calculate the quality of information $I(\mathbf{x}_{n+j})$.

Notice that in inductive conformal prediction the first three steps need only be performed once.

2.1.1 Query Functions for Active learning

A variety of query functions have been studied in the literature to the select unlabeled instances [14, 22, 28, 29, 30, 57, 58]. A brief summary of some of the most popular selection criteria is presented below.

2.1.1.1 Multiclass-level Uncertainty (MCLU)

The MCLU criterion selects the unlabeled instances that have maximum uncertainty (minimum confidence) about their correct label among all instances in the unlabeled pool. For instance, let us consider a SVM classifier. The confidence value associated with $\mathbf{x}_j$, denoted as c_j, can be computed as $c_j = d_j^{(1)} - d_j^{(2)}$ [14], where $d_j^{(1)}$ and $d_j^{(2)}$ are the largest and second largest Euclidean distances from an instance $\mathbf{x}_j$ to the separating hyperplanes, respectively.

In the CP framework, the uncertainty given by equation (2.3) is equivalent to the confidence value c_j. Several works, including [9, 12, 11], have successfully applied active learning to ICPs based on the uncertainty criterion.

2.1.1.2 Cluster Based Diversity (CBD)

Clustering techniques group similar instances into the same clusters. Since the instances within the same cluster are correlated and provide similar information, a representative instance is selected for each cluster. In [59], k-means is used to obtain a number of clusters equal to the number of instances to be selected, denoted as N_{AL}. The instance closest to each of the cluster centers is selected.

2.1.1.3 Combination of Uncertainty and Diversity

Uncertainty and diversity can be used jointly to enhance the performance of active learning [14, 22, 28].

The following optimization problem combines uncertainty and diversity in a unique query function

$$\mathbf{x}_t = \underset{\mathbf{x}_i \in T_u/T_d}{\arg\min} \left\{ \rho|c_j| + (1-\rho) \max_{\mathbf{x}_j \in T_d} S_{(\cdot)}(\mathbf{x}_i, \mathbf{x}_j) \right\}, \tag{2.4}$$

where $S_{(\cdot)}(\mathbf{x}_i, \mathbf{x}_j)$ is a similarity measure, T_d contains the set of selected instances for training (the most uncertain and diverse), T_u denotes the set containing the $L \leq |U|$ most uncertain instances, T_u/T_d represents the set of instances of T_u that are not

contained in the current set T_d, $S_{(\cdot)}(\mathbf{x}_i, \mathbf{x}_j)$ represents a similarity measure applied to instances $\mathbf{x}_i$, and $\mathbf{x}_j$, and $\rho \in [0, 1]$ provides the tradeoff between uncertainty and diversity. The first instance of T_d is selected as the most uncertain instance in T_u. The algorithm stops when the number of selected instances in T_d is equal to the number of desired instances N_{AL}.

A variety of similarity measures have been used in the literature for active learning [14, 22, 28]. Brinker *et al.* [14] use the cosine angle distance to measure the similarity between instances $\mathbf{x}_i$ and $\mathbf{x}_j$, whereas Gu *et al.* [28] employ the Gaussian kernel.

2.1.2 Generalized Batch Mode Active Learning

Chakraborty *et. al.* [26] solve the active learning problem through quadratic optimization. Let w^t be the classification rule at time t, U be the unlabeled pool, T_d be a set containing the selected unlabeled instances at time t, and define C as the possible number of classes. The relevance of an instance is given by the following expression:

$$r(T_d) = \sum_{i \in T_d} \rho_i - \lambda \sum_{i \in U/T_d} E(h|x_i, w^{t+1}), \tag{2.5}$$

where $E = -\sum_{h \in C} P(h|x_i, w^{t+1}) \log P(h|x_i, w^{t+1})$ is the entropy, U/T_d represents the set of instances of U that are not contained in the current set T_d, and λ is a tradeoff parameter. The term ρ_i is computed as the average distance of instance $\mathbf{x}_i$ to the instances in T_d. Once the relevance of all instances in U is found, the top N_{AL} most relevant instances are selected.

2.1.3 Representativeness using the Gaussian Framework

X. Li *et al.* [30] measure representativeness (information density) through entropy within a Gaussian process framework. Let $\mathcal{U}_i$ denote the index set of unlabeled instances after removing label i, and assume $\mathcal{U} = \{1, 2, \ldots, m\}$. Let $\mathcal{K}(\cdot)$ be a positive

semidefinite kernel, such as the Gaussian kernel [28]. Define the covariance matrix $\Sigma_{\mathcal{U}_i\mathcal{U}_i}$ as:

$$\Sigma_{\mathcal{U}_i\mathcal{U}_i} = \begin{bmatrix} \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i) & \dots & \mathcal{K}(\mathbf{x}_m, \mathbf{x}_i) \\ \vdots & \ddots & \vdots \\ \mathcal{K}(\mathbf{x}_i, \mathbf{x}_m) & \dots & \mathcal{K}(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}. \tag{2.6}$$

The information density for instance $\mathbf{x}_i$ can be calculated as:

$$D_{GF}(\mathbf{x}_i) = 0.5 \ln \left(-\sigma_i^2 / \sigma_{i|\mathcal{U}_i}^2 \right), \tag{2.7}$$

where $\sigma_i^2 = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_i)$, and $\sigma_{i|\mathcal{U}_i}^2 = \sigma_i^2 - \Sigma_{i\mathcal{U}_i} \Sigma_{\mathcal{U}_i\mathcal{U}_i}^{-1} \Sigma_{\mathcal{U}_i i}$. The matrix $\Sigma_{\mathcal{U}_i\mathcal{U}_i}^{-1}$ can be efficiently computed from $\Sigma_{\mathcal{U}\mathcal{U}}^{-1}$, without matrix inversion, using the algorithm described in [60].

2.1.4 Representativeness through k-Nearest Neighbors

The representativeness of an instance $\mathbf{x}_i$, denoted as d_i, in the unlabeled pool can be computed using the distance between $\mathbf{x}_i$ and its k-nearest neighbors, denoted as $\mathbf{z}_i^{(j)}$ [61], for $j = 1, \dots, k$. Define the value $\hat{d}_i$, associated with instance $\mathbf{x}_i$, as:

$$\hat{d}_i = \sum_{n=1}^{k} \left\| \mathbf{x}_i - \mathbf{z}_i^{(n)} \right\|_2^2. \tag{2.8}$$

Notice that the value $\hat{d}_i$ will be low if instance $\mathbf{x}_i$ is close to its k-nearest neighbors (densely populated region, low penalty). Conversely, the value $\hat{d}_i$ will be high if instance $\mathbf{x}_i$ is far from its k-nearest neighbors (sparsely populated region, high penalty). Let n be the size of the unlabeled pool, and let $d_{max} = \max\{d_i\}$, for $i = 1, 2, \dots, n$. The normalized representativeness of instance $\mathbf{x}_i$, denoted as d_i, can be computed as:

$$d_i = \hat{d}_i / d_{max}. \tag{2.9}$$

2.2 Distance Metric Learning (DML)

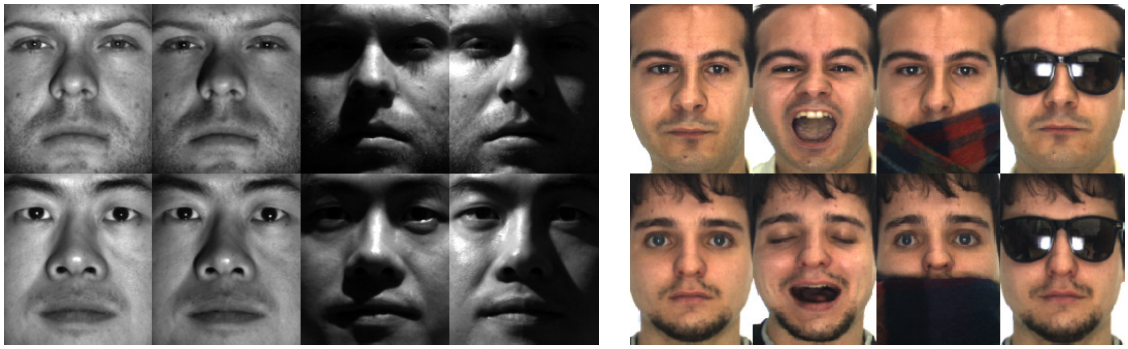
Let $\{(\mathbf{x}_i, h_i)\}_{i=1}^{n}$ be a set of n instances $\mathbf{x}_i \in \mathbb{R}^{K\times 1}$ with their corresponding class labels h_i. DML attempts to obtain a linear transformation $\mathbf{L} \in \mathbb{R}^{K\times K}$ maximizing the distances between examples belonging to different classes and minimizing the distances of examples within the same class. The original examples are then mapped onto a transformed space as $\mathbf{y}_i = \mathbf{L}\mathbf{x}_i$. Define $\mathbf{M} = \mathbf{L}^T\mathbf{L}$, the distance between two vectors can be calculated as:

$$d_M(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j), \tag{2.10}$$

where $d_M(\cdot)$ and $\mathbf{M} \in \mathbb{R}^{K\times K}$ are referred to as Mahalanobis distance and Mahalanobis matrix, respectively [37, 62]. The matrix $\mathbf{M}$ is required to be symmetric and positive semidefinite ($\mathbf{M} \succeq 0$). Bar-Hillel *et al.* [62] propose Relevant Component Analysis (RCA) to learn a Mahalanobis matrix $\mathbf{M}$. Weinberger *et al.* [37] propose to learn a Mahalanobis distance metric for k-nearest neighbor classification by semidefinite programming, referring to this approach as LMNN. The Mahalanobis matrix $\mathbf{M}$ is obtained solving the following optimization problem:

$$\begin{aligned}
&\max_{\mathbf{M}} \sum_{ij} \eta_{ij}(\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) + c\sum_{ij}\eta_{ij}(1 - h_{il})\xi_{ijl} \\
&\textbf{subject to:} \\
&1.\ (\mathbf{x}_i - \mathbf{x}_l)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_l) - (\mathbf{x}_i - \mathbf{x}_j)^T\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \geq 1 - \xi_{ijl} \\
&2.\ \xi_{ijl} \geq 0 \\
&3.\ \mathbf{M} \succeq 0,
\end{aligned} \tag{2.11}$$

where $\eta_{ij} \in \{0, 1\}$ indicate whether input $\mathbf{x}_i$ is a target neighbor of input $\mathbf{x}_j$, the binary matrix $h_{ij} \in \{0, 1\}$ indicates whether or not the labels h_i and h_j match. The hinge loss function is modeled by introducing the slack variables ξ_{ij} for all pairs of differently labeled inputs, *i.e.*, for all $\langle i, j\rangle$ such that $y_{ij} = 0$. The LMNN solver is based on a



(a) (b)

Figure 2.1: Images from (a) Extended YaleB database and (b) AR database.

combination of sub-gradient descent in both the matrices $\mathbf{L}$ and $\mathbf{M}$, the latter used mainly to verify convergence. The updates in $\mathbf{M}$ are projected back onto the positive semidefinite cone after each step [37].

2.3 Image Database Description

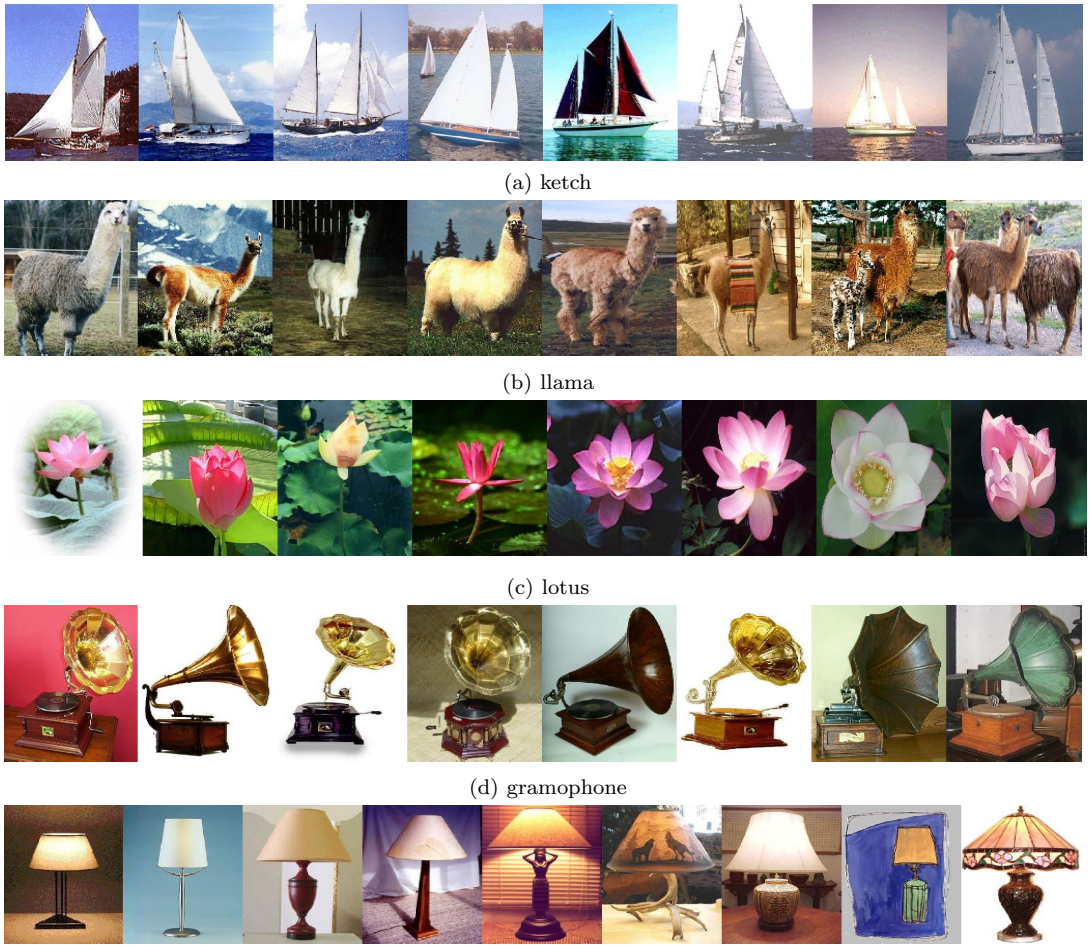
Two face databases are considered in this work, the Extended YaleB database [63] and the AR face database [64], along with one object recognition database, Caltech101 [65], and one emotion recognition database, Oulu-CASIA NIR&VIS facial expression [66]. We describe each one of the databases below.

2.3.1 Extended YaleB Database

The Extended YaleB database consists of 2,414 frontal-face images of 38 people (38 different classes) taken under varying lightning conditions. There are about 64 images for each person. The images are cropped to 192×168 pixels and normalized. Example images from the Extended YaleB database are shown in Fig. 2.1(a).

2.3.2 AR Database

The AR database contains over 4,000 frontal-face images of 100 people (100 different classes), each of size 165×120. Compared to the Extended Yale B database, these images include more facial variations and also facial disguises, such as sunglasses



(a) ketch

(b) llama

(c) lotus

(d) gramophone

(e) lamp

Figure 2.2: Images from the Caltech101 database.

and scarves, which makes classification more challenging. Example images from the AR database are shown in Fig. 2.1(b).

2.3.3 Caltech101

The Caltech101 dataset [65] contains 9,144 images from 102 classes (101 object classes and a background class) including animals, vehicles, flowers, etc. The samples within the same category display considerable shape variability. The number of images in each category varies from 31 to 800. Example images from the Caltech101 database are shown in Fig. 2.2.

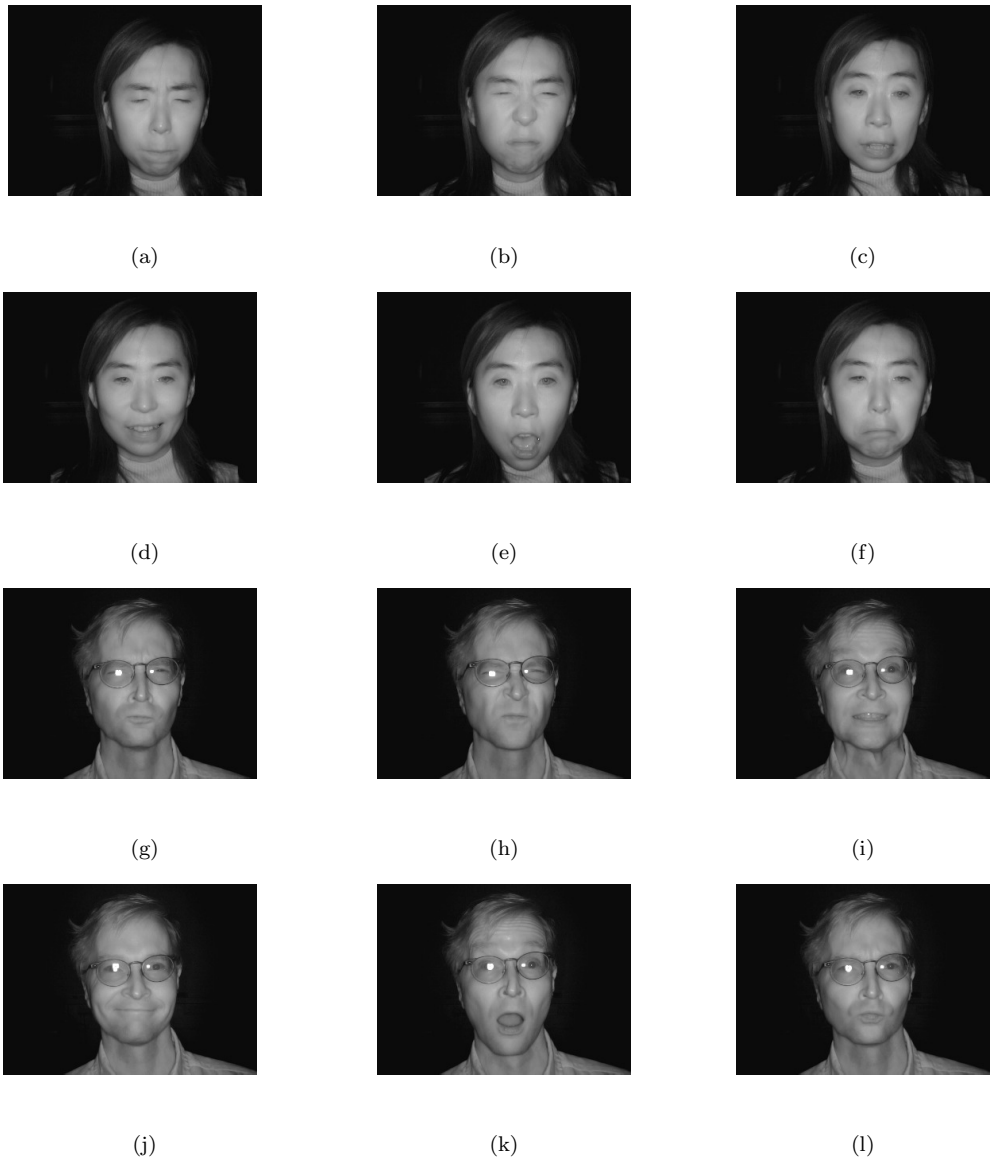


Figure 2.3: Images from the Oulu-CASIA database. Subject 2: (a) angry, (b) disgust, (c) fear, (d) happy, (e) surprise, (f) sad. Subject: 38 (g) angry, (h) disgust, (i) fear, (j) happy, (k) surprise, (l) sad.

2.3.4 Oulu-CASIA NIR&VIS database

The Oulu-CASIA NIR&VIS facial expression database [66] consists of six expressions (surprise, happiness, sadness, anger, fear, and disgust) from 80 people between 23 to 58 years old. The imaging hardware works at rate of 25 frames per second and

image resolution is 320×240 pixels. All expressions are captured in three different illumination conditions: normal, weak and dark. The number of video sequences is 480 (80 subjects by six expressions) for each illumination and imaging system pair, so totally there are 2880 (480×6) video sequences in the database. Example images extracted from the video snippets are shown in Fig. 2.3.

Chapter 3

CONFORMAL PREDICTION BASED ACTIVE LEARNING FOR SPARSE CODING CLASSIFIERS

Two types of DL approaches are considered in this work: synthesis dictionary learning (SDL), and dictionary pair learning (DPL). The two aforementioned techniques are briefly introduced in this section. For the following definitions, let $\mathbf{Y} \in \mathbb{R}^{N \times n}$ be a matrix composed of n training vectors $\mathbf{y} \in \mathbb{R}^{N \times 1}$, $\mathbf{X} \in \mathbb{R}^{K \times n}$ be a matrix composed of vectors $\mathbf{x} \in \mathbb{R}^{K \times 1}$, which are the sparse representations of the training vectors in matrix $\mathbf{Y}$, and M be the number of classes. Let $\mathbf{D} \in \mathbb{R}^{N \times K}$ be the dictionary, constituted by K atoms $\mathbf{d} \in \mathbb{R}^{N \times 1}$ that are the columns of $\mathbf{D}$.

3.1 Synthesis Dictionary Learning

A reconstructive dictionary $\mathbf{D} \in \mathbb{R}^{N \times K}$ is learned by solving $< \mathbf{X}, \mathbf{D} >= \arg\min_{\mathbf{X},\mathbf{D}} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2$. This optimization problem is solved by alternating between the updates of $\mathbf{D}$ and $\mathbf{X}$ [67]. LC-KSVD [43] and LC-RLSDLA [51] use an augmented version of matrix $\mathbf{Y}$, including the class label information, to simultaneously obtain a linear classifier $\mathbf{W} \in \mathbb{R}^{M \times K}$. Define $\mathbf{u} = [u_1, \ldots, u_M]^T = \mathbf{W}\mathbf{x}$. The predicted label for $\mathbf{x}$ is obtained as $\hat{h} = \arg\max_j u_j$, for $j = 1, \ldots, M$.

3.2 Dictionary Pair Learning

DPL [68, 47] learns M synthesis dictionaries $\mathbf{D}_j \in \mathbb{R}^{N \times K}$, and M analysis dictionaries $\mathbf{P}_j \in \mathbb{R}^{K \times N}$ $(j = 1, \ldots, M)$. DPL solves the following optimization problem: $< \mathbf{P}, \mathbf{D} >= \arg\min_{\mathbf{P},\mathbf{D}} \sum_{j=1}^{M} \left\|\mathbf{Y}_j - \mathbf{D}_j\mathbf{P}_j\mathbf{Y}_j\right\|_F^2 + \left\|\mathbf{P}_j\bar{\mathbf{Y}}_j\right\|_2^2$, where $\mathbf{Y}_j$ is a matrix containing the training vectors of class j, and $\bar{\mathbf{Y}}_j$ is a the complementary data matrix of $\mathbf{Y}_j$. Define $v_j = \left\|\mathbf{x} - \mathbf{D}_j\mathbf{P}_j\mathbf{x}\right\|_2$. The predicted label for $\mathbf{x}$ is obtained as $\hat{h} = \arg\min_j v_j$, for $j = 1, \ldots, M$.

3.3 CPAL-SCC: Conformal Prediction Based Active Learning for Sparse Coding Classifiers

We propose an active learning algorithm within de CP framework, referred to as CPAL-SCC, in which instances are selected from an unlabeled pool based on two criteria, uncertainty and diversity. In the remainder of this section the proposed nonconformity measures and query function are introduced, and the CPAL-SCC algorithm is described.

3.3.1 CPAL-SCC Nonconformity Measures

We propose two nonconformity measures, the first one is designed for SDL, and the second one for DPL. A description of the nonconformity measures is provided below.

3.3.1.1 Nonconformity measure for SDL

Let $\mathbf{W} \in \mathbb{R}^{M\times K}$ be a linear classifier, for M distinct class labels, constituted by row vectors $\mathbf{w}_q \in \mathbb{R}^K$, $q \in \{1, 2, \ldots, M\}$. Define $\hat{\mathbf{w}}_q = \mathbf{w}_q / \left\|\mathbf{w}_q\right\|$. The proposed nonconformity measure for SDL under the null hypothesis $\mathcal{H}_q$ is given by

$$A_{SDL}^{(\mathcal{H}_q)} := -\hat{\mathbf{w}}_q \mathbf{x} + \frac{1}{M-1}\sum_{i\neq q} \hat{\mathbf{w}}_i \mathbf{x}, \tag{3.1}$$

3.3.1.2 Nonconformity measure for DPL

Let $\mathbf{D}_j \in \mathbb{R}^{N\times K}$, and $\mathbf{P}_j \in \mathbb{R}^{K\times N}$ be the synthesis and analysis dictionaries for class j ($j = 1, \ldots, M$), respectively. The proposed nonconformity measure for DPL under the null hypothesis $\mathcal{H}_q$ is given by

$$A_{DPL}^{(\mathcal{H}_q)} := \left\|\mathbf{x} - \mathbf{D}_q\mathbf{P}_q\mathbf{x}\right\|_2 - \frac{1}{M-1}\sum_{i\neq q} \left\|\mathbf{x} - \mathbf{D}_i\mathbf{P}_i\mathbf{x}\right\|_2, \tag{3.2}$$

Assuming that the classifiers are accurate and the null hypothesis $\mathcal{H}_q$ is true, the values of $A_{SDL}^{(\mathcal{H}_q)}$, and $A_{DPL}^{(\mathcal{H}_q)}$ will decrease (it may become negative for SDL), indicating

Algorithm 1 CPAL-SCC

1: **Input:** Proper training set $T_{prop} = \{z_1, ..., z_\ell\}$, calibration set $T_{cal} = \{z_{\ell+1}, ..., z_{\ell+r}\}$, unlabeled pool $U = \{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+v}\}$, classification rule C_{prop}, number of desired instances N_{AL}, and number of class labels M

2: Use Equation (3.1) or (3.2) and the classification rule C_{prop} to calculate:

- The nonconformity scores $\{\alpha_{\ell+1}, \ldots, \alpha_{\ell+r}\}$ corresponding to the instances in the calibration set.
- The nonconformity scores $\{\alpha_{n+1}^{\mathcal{H}_i}, \ldots, \alpha_{n+v}^{\mathcal{H}_i}\}$ corresponding to the instances in the unlabeled pool, where $i = \{1, \ldots, M\}$

3: Use Equation (2.1) to calculate the p-values associated with the instances in U, and obtain their confidence $s(\mathbf{x}_{n+j})$ through equation (2.2), where $j \in \{1, \ldots, v\}$

4: Apply equation (3.3) to select N_{AL} most uncertain and diverse instances in the unlabeled pool. Then group such instances and their corresponding class labels as $T_d = \{z_1^d, \ldots, z_{N_{AL}}^d\}$

5: Construct $T_{AL} = T_{prop} \cup T_d$

6: **Output:** T_{AL}

that $\mathbf{x}$ conforms to class q. Conversely, if the null hypothesis is false, the value of $A_{SDL}^{(\mathcal{H}_q)}$, and $A_{LM}^{(\mathcal{H}_q)}$ will tend increase, indicating that $\mathbf{x}$ does not conform to that particular class.

3.3.2 CPAL-SCC Query Function

Different from previous work on ICP [12, 13], the proposed approach considers both uncertainty and diversity as the selection criteria for active learning. The proposed query function is given by

$$\mathbf{x}_t = \underset{\mathbf{x}_i \in T_i/T_d}{\arg\min} \left\{ \rho s(\mathbf{x}_i) + (1-\rho) \max_{\mathbf{x}_j \in T_d} \left[\frac{|\mathbf{x}_i \cdot \mathbf{x}_j|}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \right] \right\}, \tag{3.3}$$

where U, T_d and, U/T_d are the sets containing the instances in U, the instances selected for training, and the instances in U that are not contained in T_d, respectively. Diversity is measured by the second term using cosine angle distance [14]. The parameter ρ provides the trade-off between uncertainty and diversity. The first instance of T_d is selected as the instance with the highest uncertainty in T_i. The algorithm stops when the number of selected instances in T_d is equal to the desired number N_{AL}.

3.3.3 CPAL-SCC Algorithm

CPAL-SCC selects the most uncertain and diverse instances from an unlabeled pool using the proposed query function described in (3.3). The selected instances,

along with their corresponding class labels, are used in a subsequent training stage to improve performance, instead of relying on instances that are selected at random.

Define $T_{train} = \{z_1, ..., z_n\}$ as the training set and $U = \{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+v}\}$ as the unlabeled pool. Following the steps described for ICP in [1], we split T_{train} into $T_{prop} = \{z_1, ..., z_\ell\}$, the proper training set, and $T_{cal} = \{z_{\ell+1}, ..., z_{\ell+r}\}$, the calibration set, where the size of the training set satisfies $n = \ell + r$. Let C_{prop} be the classification rule obtained through the underlying algorithm employing only T_{prop}. Let N_{AL} and M be the number of desired instances from U and the number of class labels, respectively. Let $T_{AL} = T_{prop} \cup T_d$, where $T_d = \{z_1^s, \ldots, z_{N_{AL}}^s\}$ is the set of pairs containing the N_{AL} most uncertain and diverse instances in U and their corresponding class labels. The proposed active learning approach is summarized in Algorithm 1.

3.4 Experimental Results

The focus of CPAL-SCC is twofold: 1) to improve the performance of sparse coding classifiers through active learning; and 2) to produce reliable confidence values. Therefore, CPAL-SCC is to evaluated based on the improvement achieved in classification performance and the quality of the produced confidence values.

3.4.1 Experimental Setup

The performance of CPAL-SCC is evaluated using two different sparse coding algorithms: LC-KSVD [43], and DPL [68]. The baseline for our experiments is random sampling. Experiments are conducted on the Extended YaleB, AR, and Caltech 101 databases. The feature descriptors are used for the Extended YaleB and AR databases are randomfaces of size $N = 504$ and $N = 540$, respectively. For Caltech101, SIFT descriptors are first extracted. Next, spatial pyramid features, based on the SIFT descriptors, are obtained. The dimension of the spatial pyramid features is then reduced to 3000 through PCA [43].

For each of the experiments, 5 trials are conducted. In each trial, the order of the training instances is permuted. The average classification accuracy is presented.

The number of images per class in the proper training set for the Extended YaleB, AR, and Caltech101 databases is 8, 5, and 5, respectively. The calibration set consists of 199 instances, which results in a resolution of 0.5% in the confidence values, according to (2.1). Optimization is performed over the parameter ρ through exhaustive search, and the best results are presented.

3.4.2 Results: CPAL-SCC for Active Learning

The performance improvement obtained through CPAL-SCC is compared with that of: random sampling, active learning based on uncertainty [9, 12, 13], and MCLU-ECDB [22], which are denoted as (rnd), AL(MCLU), and AL(MCLU-ECDB) respectively. The performance of LC-KSVD and DPL as a function of the number of selected instances N_{AL}, for the different databases and query functions, is shown in Fig. 3.1, 3.2, and 3.3. It is observed that the performance of both algorithms is improved when CPAL-SCC is used, for all the considered databases.

Table 3.1 shows that for the Extended YaleB database (DPL, $N_{AL} = 200$) the performance of (rnd), AL(MCLU), and AL(MCLU-ECDB) is 92.3%, 95.8%, and 96.2%, respectively, whereas that of CPAL-SCC is 96.9%.

For the AR database (LC-KSVD, $N_{AL} = 300$), it is observed in Table 3.1 that the classification accuracy of (rnd), AL(MCLU), and AL(MCLU-ECDB) is 74.0%, 77.9%, and 78.1%, respectively, whereas that of CPAL-SCC is 79.9%.

Similarly, for Caltech101 (DPL, $N_{AL} = 200$), Table 3.1 shows that for DPL, the classification accuracy of (rnd), AL(MCLU), and AL(MCLU-ECDB) is 50.8%, 51.1%, and 51.5%, respectively, whereas that of CPAL-SCC is 52.5%.

The effect of the parameter ρ on the performance of LC-KSVD (AR database) is shown in Fig. 3.4. Notice that ρ has to be optimized for the each value of N_{AL}. Similar results are obtained for the YaleB, and Caltech101 databases.

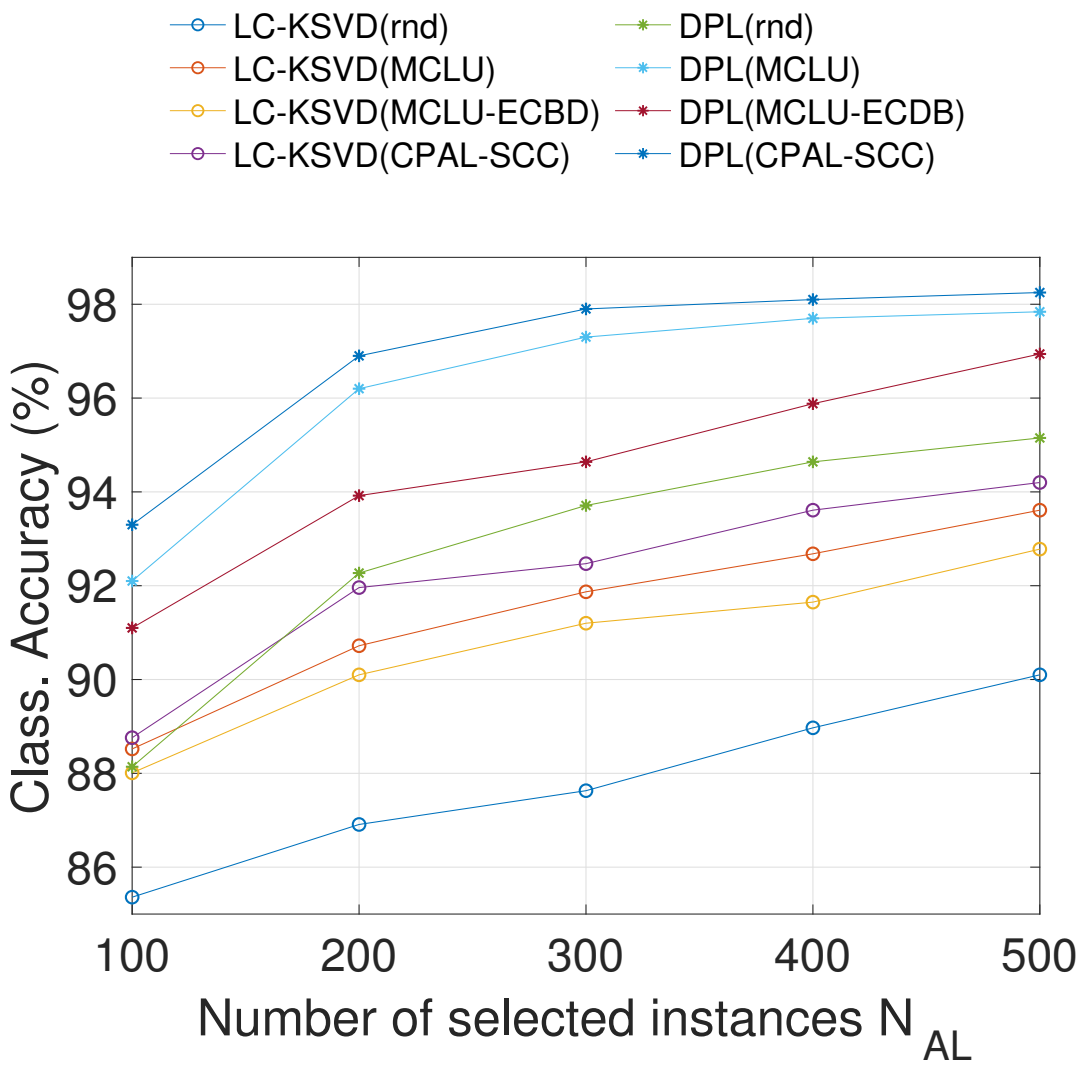


Figure 3.1: Classification accuracy (%) for DPL and LC-KSVD as a function of N_{AL}, YaleB ($K = 380$).

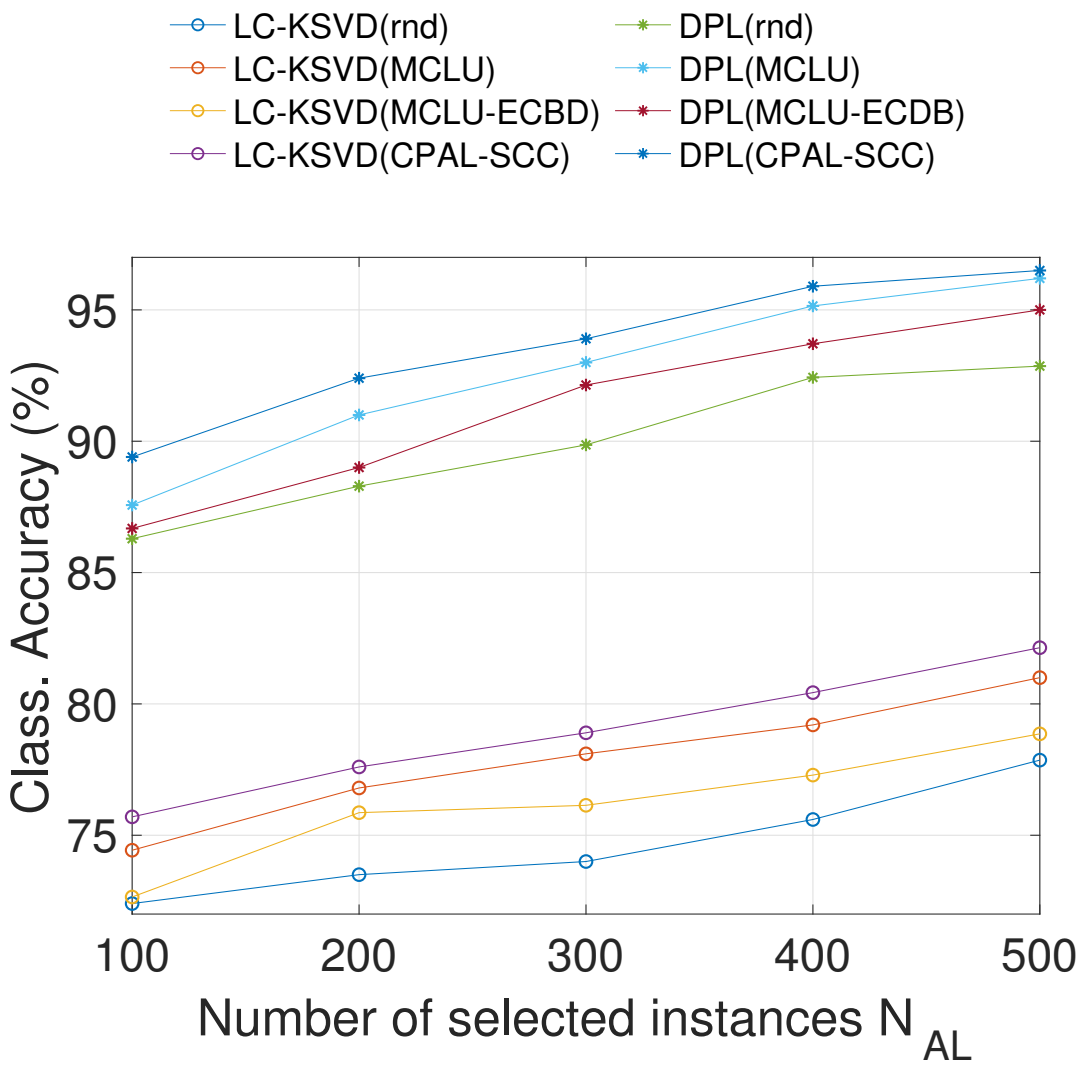


Figure 3.2: Classification accuracy (%) for DPL and LC-KSVD as a function of N_{AL}, AR ($K = 400$).

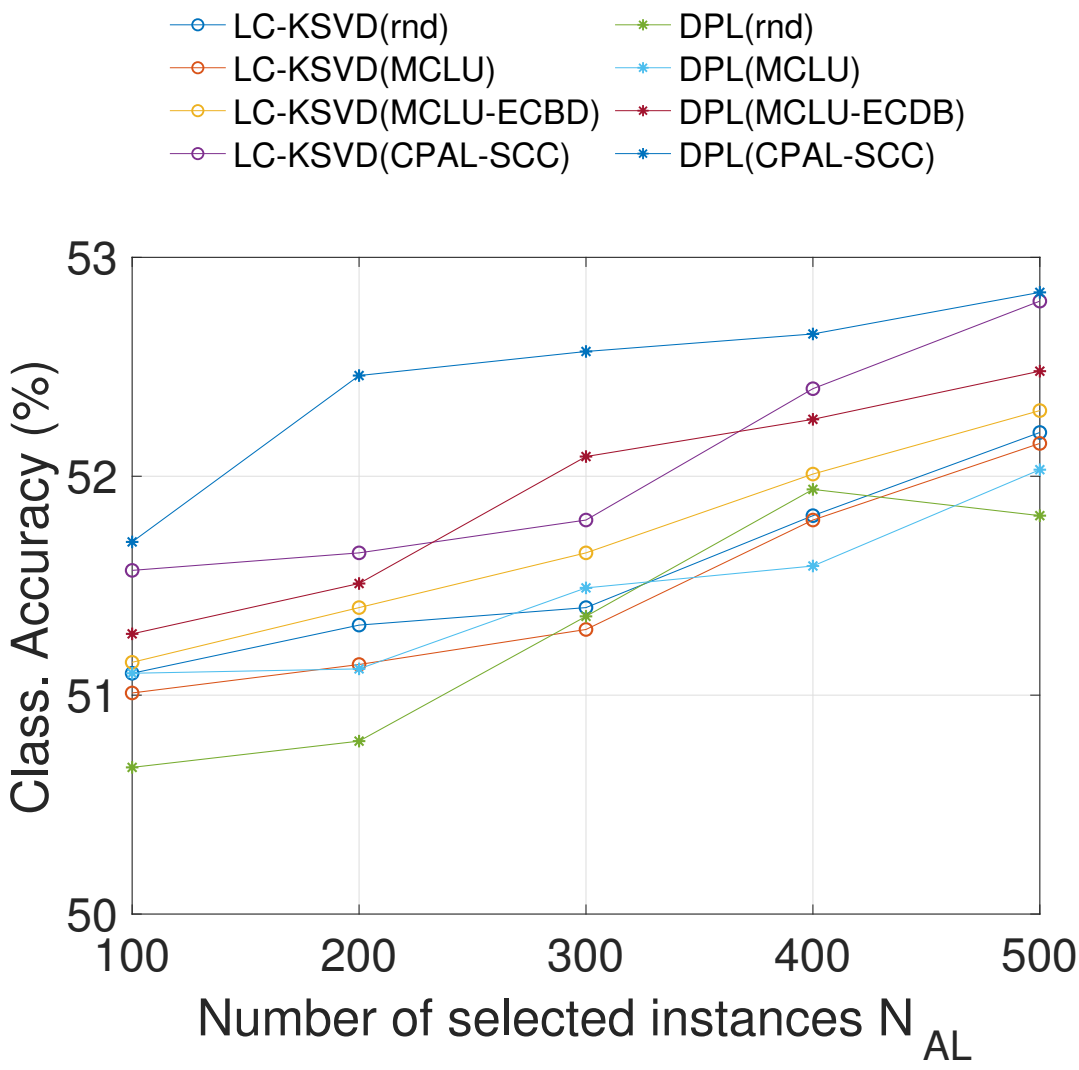


Figure 3.3: Classification accuracy (%) for DPL and LC-KSVD as a function of N_{AL}, Caltech101 ($K = 510$).

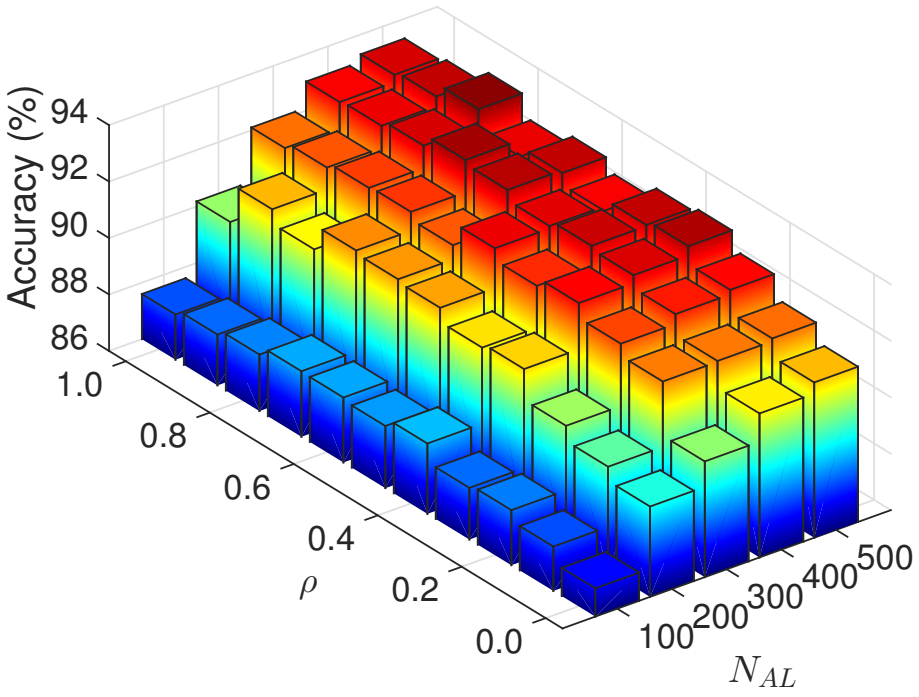


Figure 3.4: Effect of ρ on the performance of LC-KSVD (AR).

Table 3.1: Classification accuracy (%) for different query functions as a function of the number of selected instances N_{AL}.

Algorithm	Query func.	YaleB		AR		Cal101	
		N_{AL}		N_{AL}		N_{AL}	
		200	300	200	300	200	300
LC-KSVD	(rnd)	86.9	87.6	73.5	74.0	51.3	51.4
	AL(MCLU)	90.1	91.4	76.5	77.9	51.1	51.3
	AL(MCLU-ECBD)	90.7	91.8	76.8	78.1	51.4	51.6
	CPAL-SCC	**91.9**	**92.5**	**77.6**	**79.9**	**51.7**	**51.8**
DPL	(rnd)	92.3	93.7	88.3	89.9	50.8	51.4
	AL(MCLU)	95.8	97.3	91.0	92.4	51.1	51.5
	AL(MCLU-ECBD)	96.2	97.4	91.3	92.9	51.5	52.1
	CPAL-SCC	**96.9**	**97.9**	**92.4**	**93.9**	**52.5**	**52.6**

Table 3.2: Experimental results of the validity property.

Algorithm	Confidence (%)	Error (%)		
		YaleB	AR	Cal101
LC-KSVD	95	4.8	4.7	3.9
	90	10.0	12.2	9.9
	85	15.5	15.9	14.3
DPL	95	5.1	5.7	4.2
	90	10.4	11.7	9.2
	85	15.8	16.2	15.6

3.5 Chapter Conclusion

In this chapter we propose a conformal prediction based active learning algorithm for sparse coding classifiers, referred to as CPAL-SCC, which considers uncertainty and diversity as the selection criteria. Moreover, two nonconformity measures, one for synthesis dictionary learning, and the other one for dictionary pair learning are proposed. Experiments conducted on face and object recognition databases demonstrate that CPAL-SCC improves the classification accuracy of state-of-the-art dictionary learning algorithms, while producing reliable confidence values.

In the following chapter the concepts of distance metric learning and representativeness are introduced to further improve the performance of active learning.

Moreover, a nonconformity measure for convolutional neural networks is presented.

Chapter 4

CONFORMAL PREDICTION BASED ACTIVE LEARNING FOR CONVOLUTIONAL NEURAL NETWORKS

We propose an active learning algorithm within de CP framework, referred to as CPAL-CNN, which uses a novel nonconformity measure that produces reliable confidence values. CPAL-CNN selects instances from an unlabeled pool based on the evaluation of three criteria: uncertainty, diversity, and representativeness. In the remainder of this section the proposed nonconformity measure and query function are introduced, and the CPAL-CNN algorithm is described.

4.1 CPAL-CNN Nonconformity Measure

Consider a CNN with M ouputs, corresponding to M different class labels. Let $\mathbf{x}_j$ be an input instance, and $h_j \in \{1, \dots, M\}$ be its corresponding class label ($j = 1, 2, \dots$). Let the outputs of the CNN satisfy $o_j^{(i)} = P(h_j = i|\mathbf{x}_j, \boldsymbol{\Theta})$, where $\boldsymbol{\Theta}$ represents the parameters of the CNN, *i.e.*, the predicted class is given by the expresion $\max_{i=1,\dots,M} o_j^{(i)}$, and the outputs satisfy $\sum_{i=1}^{M} o_j^{(i)} = 1$. The proposed nonconformity measure is given by:

$$A_{CNN}^{(\mathcal{H}_q)} := 1 - \gamma o_j^{(q)} + (1 - \gamma) \max_{i=1,\dots,M, i \neq q} o_j^{(i)}, \tag{4.1}$$

where $A_{CNN}^{(\mathcal{H}_q)}$ represents the proposed nonconformity measure under the null hypothesis $\mathcal{H}_q$. The second term in Equation (4.1) represents the q-th output of the CNN for an input instance $\mathbf{x}_j$ ($j = 1, 2, \dots$). The third term represents the output with the highest value, different from the q-th output, taken from the remaining $M-1$ outputs. Assuming that the CNN is accurate and the null hypothesis $\mathcal{H}_q$ is true, the second

term in equation (4.1) will be large, preceded by a negative sign, outweighing the positive third term, meaning that instance $\mathbf{x}_j$ conforms to class q. Conversely, if the null hypothesis is false, the third term in (4.1) will tend to outweigh the second term, indicating that $\mathbf{x}_j$ does not conform to that particular class. The term $\gamma \in [0, 1]$ is introduced to provide a tradeoff between the importance of the second and third terms. Notice that the hinge and margin nonconformity measures are special cases of (4.1), when $\gamma = 1$ and $\gamma = 0.5$, respectively.

4.2 CPAL-CNN Query Function

The proposed query function selects instances from the unlabeled pool based on the evaluation of three criteria: uncertainty, diversity, and representativeness (information density). Different from previous work on active learning, the proposed query function measures diversity and information density in a reduced space, obtained through Principal Component Analysis (PCA), thereby reducing the computational burden and allowing the use of DML techniques [62, 37]. In particular, LMNN [37] is utilized to obtain the Mahalanobis matrix $\mathbf{M}$ that adapts to the statistics of the database being used, which is then used to measure the similarity between different instances. Moreover, representativeness is considered to remove possible outliers in the selection process. We adapt the query function given by equation (2.4) to the CP framework. The term c_j is replaced by the confidence $s(\cdot)$, which is defined in equation (2.2), and two more terms, associated with diversity and representativeness, are also considered. Let $\tilde{\mathbf{x}}_i$ denote the low-dimensional representation of instance $\mathbf{x}_i$, obtained through PCA. The proposed query function is given by

$$\mathbf{x}_t = \underset{\mathbf{x}_i \in U/T_d}{\arg\min} \left\{ (1 - \alpha - \beta) s(\mathbf{x}_i) + \alpha \max_{\mathbf{x}_j \in T_d} S_M(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) - \beta D(\tilde{\mathbf{x}}_i) \right\}, \tag{4.2}$$

where $S_M(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ measures the similarity between the low-dimensional vectors $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ using the Mahalanobis matrix $\mathbf{M}$, obtained through LMNN. The term $S_M(\mathbf{x}_i, \mathbf{x}_j)$

Algorithm 2 CPAL-CNN

1: **Input:** Proper training set $T_{prop} = \{z_1, ..., z_\ell\}$, calibration set $T_{cal} = \{z_{\ell+1}, ..., z_{\ell+r}\}$, unlabeled pool $U = \{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+v}\}$, CNN parameters $\boldsymbol{\Theta}_{prop}$, number of desired instances N_{AL}, and number of class labels M

2: Perform PCA to obtain the low-dimensional representations of T_{prop} and U, denoted as $\tilde{T}_{prop}$ and $\tilde{U}$, respectively

3: Compute information density for the instances in $\tilde{U}$, as described in equation (2.7)

4: Use LMNN to obtain the Mahalanobis matrix $\mathbf{M}$, employing the low-dimensional training set $\tilde{T}_M = \tilde{T}_{prop} \cup \tilde{T}_D$ (DML is performed on a set containing instances that are representative of the data)

5: Use equation (4.1) along with the parameters $\boldsymbol{\Theta}_{prop}$ to calculate:

- The nonconformity scores $\{\alpha_{\ell+1}, \ldots, \alpha_{\ell+r}\}$ corresponding to the instances in the calibration set.
- The nonconformity scores $\{\alpha_{n+1}^{\mathcal{H}_i}, \ldots, \alpha_{n+v}^{\mathcal{H}_i}\}$ corresponding to the instances in the unlabeled pool, where $i = \{1, \ldots, M\}$

6: Use Equation (2.1) to calculate the p-values associated with the instances in U, and obtain their confidence $s(\mathbf{x}_{n+j})$ through equation (2.2), where $j \in \{1, \ldots, v\}$

7: Apply equation (4.2) to select N_{AL} instances based on uncertainty, diversity, and information density. Then group those instances and their corresponding labels as $T_d = \{z_1^d, \ldots, z_{N_{AL}}^d\}$

8: Construct $T_{AL} = T_{prop} \cup\ T_d$

9: **Output:** T_{AL}

is defined as

$$S_M(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}(\mathbf{x}_i - \mathbf{x}_j)}{2\sigma}\right). \tag{4.3}$$

where σ is a hyperparameter to be optimized. The terms U, T_d and, U/T_d in (4.2) denote the unlabeled pool, the current set of instances selected for training, and the set of instances of U that are not contained in the current set T_d, respectively. The parameters $\{\alpha, \beta \in [0, 1] \mid \alpha + \beta \leq 1\}$ provide a tradeoff between uncertainty, diversity, and representativeness. Representativeness is calculated using the Gaussian framework, as described by equation (2.7). The first instance of T_d is selected as the most informative instance in U. The algorithm stops when the number of selected instances in T_d is equal to N_{AL}.

4.3 CPAL-CNN Algorithm

We propose an active learning algorithm for convolutional neural networks within the CP framework. First, we split the training set into the proper training set, and the calibration set, as described in Section 2. Then, the nonconformity scores of the instances in calibration set and the unlabeled pool are computed using equation (4.1). The nonconformity scores are used to compute the p-values and the confidence of the instances in the unlabeled pool according to equation (2.1) and (2.2), respectively. The low-dimensional representation of the instances in the unlabeled pool is obtained through PCA, and the information density is measured within the Gaussian framework, using equation (2.7). DML is performed in the reduced space to obtain the Mahalanobis matrix $\mathbf{M}$. Dimensionality reduction, DML, and the computation of information density need only be performed once at the beginning of the algorithm. Then, instances are selected from the unlabeled pool through the query function described by equation (4.2), which considers uncertainty, diversity, and information density. We derive the active learning mode for the offline setting, meaning that the entire unlabeled pool is used as a batch.

Define $T_{train} = \{z_1, ..., z_n\}$ as the training set, $U = \{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+v}\}$ as the unlabeled pool. Following the steps described for ICPs in Section 2, we split T_{train} into $T_{prop} = \{z_1, ..., z_\ell\}$, the proper training set, and $T_{cal} = \{z_{\ell+1}, ..., z_{\ell+r}\}$, the calibration set, where the size of the training set satisfies $n = \ell + r$. Let $\tilde{T}_{prop}$ and $\tilde{U}$ be the low-dimensional representation of the sets T_{prop} and U obtained through PCA, respectively. Let $\boldsymbol{\Theta}_{prop}$ represent the parameters of the CNN obtained employing T_{prop} for training. Let N_{AL} and M be the number of desired instances from U and the number of class labels, respectively. Define $\tilde{T}_M = \tilde{T}_{prop} \cup \tilde{T}_D$, where $\tilde{T}_D$ is the set of pairs containing the N_{AL} instances with the highest information density from $\tilde{U}$, *i.e.*, the instances in $\tilde{T}_D$ are representative of the data and do not lie in sparsely populated regions (outliers). Define $T_{AL} = T_{prop} \cup \ T_d$, where $T_d = \{z_1^s, \ldots, z_{N_{AL}}^s\}$ is the set of pairs containing the N_{AL} instances from U selected through the query function described in (4.2), and their corresponding class labels. Define $T_R = T_{prop} \cup T_{rnd}$, where T_{rnd} is a set containing

Table 4.1: CNN architecture for the Extended YaleB database.

Layers	Filter Size	Stride	Padding	Output $W \times H \times L$
Input	-	-	-	$32 \times 32 \times 1$
Conv-ReLU Avg_pool	5×5 2×2	1 2	0	$14 \times 14 \times 25$
Conv-ReLU Avg_pool	5×5 2×2	1 2	0	$10 \times 10 \times 65$
FC-ReLU Dropout	-	-	-	400
FC-Softmax	-	-	-	38

N_{AL} pairs of instances with their corresponding class labels selected at random from the unlabeled pool.

The performance of the underlying algorithm can be improved employing T_{AL}, instead of T_R, in a subsequent training stage. CPAL-CNN returns the training set T_{AL}. The proposed approach is summarized in Algorithm 2.

4.4 Experimental Results

The focus of CPAL-CNN is twofold: 1) to improve the performance of CNNs through active learning; and 2) to produce reliable confidence values. Therefore, our goal is to evaluate CPAL-CNN based on the improvement achieved in classification performance and the quality of the produced confidence values. This section is organized as follows. First, we present the experimental setup and provide a description of the databases used in this paper; second, we present experimental results to evaluate the performance of CPAL-CNN for active learning, providing a comparison between the proposed technique and previous work. Last, we demonstrate the quality of the confidence values obtained through CPAL-CNN.

4.4.1 Experimental Setup

Experiments are conducted on two face databases, the Extended YaleB database [63] and the AR face database [64], and one object recognition database, Caltech101 [65].

Table 4.2: CNN architecture for the AR database.

Layers	Filter Size	Stride	Padding	Output $W \times H \times L$
Input	-	-	-	$50 \times 50 \times 1$
Conv-ReLU	7×7	1	0	$22 \times 22 \times 15$
Avg_pool	2×2	2		
Conv-ReLU	7×7	1	0	$8 \times 8 \times 45$
Avg_pool	2×2	2		
FC-ReLU Dropout	-	-	-	500
FC-Softmax	-	-	-	100

Table 4.3: CNN architecture for the Caltech101 database.

Layers	Filter Size	Stride	Padding	Output $W \times H \times L$
Input	-	-	-	$32 \times 32 \times 3$
Conv-ReLU	5×5	1	2	$32 \times 32 \times 32$
Avg_pool	3×3	2	1 bot/right	
Conv-ReLU	5×5	1	2	$32 \times 32 \times 32$
Avg_pool	3×3	2	1 bot/right	
Conv-ReLU	5×5	1	2	$32 \times 32 \times 64$
Avg_pool	3×3	2	1 bot/right	
FC-ReLU Dropout	-	-	-	400
FC-Softmax	-	-	-	101

The performance of CPAL-CNN is evaluated over three different CNNs architectures, one for each database. The reference classification accuracy is taken as that obtained when T_R is employed for training, *i.e.*, instances are selected randomly from the unlabeled pool U. CPAL-CNN is implemented in conjunction with the different CNN architectures to produce the training set T_{AL}, which is then used to improve performance. The quality of the CPAL-CNN confidence values is demonstrated through the evaluation of validity property [12] across the different databases.

For each of the experiments in this section, 5 trials are conducted. In each trial, the order of the instances in the training set is permuted. The average classification accuracy is presented. The proper training set T_{prop} consists of 10 images per class

for the Extended YaleB database, 5 images per class for the AR database, and 10 images per class for Caltech101. The calibration set consists of 199 instances for all the experiments, which results in a resolution of 0.5% in the confidence values calculated, according to equation (2.1).

Experiments are conducted on two face databases, the Extended YaleB database [63] and the AR face database [64], and one object recognition database, Caltech101 [65]. Example images from the Extended YaleB database are shown in Fig. 2.1(a). The original images are reshaped to 32×32 pixels. The CNN architecture used for this database is described in Table 4.1. Example images from the AR database are shown in Fig. 2.1(b). As part of the preprocessing, the orginal images are converted to greyscale and reshaped to 50×50 pixels. The CNN architecture used for this database is described in Table 4.2. Example images from the Caltech101 database are shown in Fig. 2.2. The original images are reshaped to $32 \times 32 \times 3$ pixels (RGB format). The CNN architecture used for this database is described in Table 4.3.

4.4.2 Results: CPAL-CNN for Active Learning

In this set of experiments, CPAL-CNN is implemented in conjunction with three different CNN architectures, one for each database. The performance is presented as a function of the number of selected instances, N_{AL}, from the unlabeled pool U. We compare the performance improvement obtained through CPAL-CNN with that of the following approaches: random sampling, *i.e.*, we take instances from the unlabeled pool at random, active learning based only on uncertainty [9, 12, 13], active learning considering uncertainty and ABD [14], and active learning considering uncertainty and KBD [28], which are denoted as (rnd), AL(MCLU), AL(MCLU-ABD), and AL(MCLU-KBD), respectively. Random sampling is used as the baseline for the experiments. Parameter optimization using exhaustive search is performed over the weights, α, β and the kernel variance, σ, for the proposed query function. For AL(MCLU-ABD) and AL(MCLU-KBD), the parameters ρ, L, and σ are optimized using the same approach. The best results are presented.

For random sampling, the training set $T_R = T_{prop} \cup T_{rnd}$ is employed, where T_{rnd} contains N_{AL} randomly selected instances from U with their corresponding class labels, and T_{prop} is the proper training set. The results for active learning are obtained using the training set $T_{AL} = T_{prop} \cup T_d$, where T_d contains N_{AL} instances selected from U using the aforementioned active learning approaches, with their corresponding class labels. The size of the unlabeled pool $|U|$ for the different databases is the following: for the Extended YaleB database $|U| = 912$, for the AR database we select $|U| = 1500$, and for Caltech101 $|U| = 3264$. The parameter γ is set to 1 in the nonconformity score given by (4.1). Diversity and information density are measured in a reduced space, obtained through PCA. The dimensionality is reduced until 95% of the variance is explained. The size of the instances in the reduced space for the Extended YaleB, AR, and Caltech101 databases is 59×1, 76×1, and 214×1, respectively.

The performance of CPAL-CNN as a function of the number of selected instances N_{AL}, for the different databases and query functions, is shown in Fig. 4.1, 4.2, and 4.3. It is observed that the classification accuracy of the different CNN architectures is improved significantly when active learning is used, for all the considered databases. Notice that when uncertainty, diversity, and information density are taken into account, the performance is improved. This occurs since similar instances are not selected and outliers are rejected. Moreover, it is observed that the performance of CPAL-CNN is the best among all the considered approaches. This demonstrates the effectiveness of the proposed approach.

The results for the Extended YaleB database in Fig. 4.1 show that the biggest performance gain is obtained for $N_{AL} = 400$, and the performance gain due to active learning stays within about 8.2%, using random sampling as the baseline. Table 4.4 shows that for $N_{AL} = 400$ the performance of random sampling is 74.8%, whereas that of CPAL-CNN is 83.0%.

The results for the AR database in Fig. 4.2 show that the largest performance gain is obtained when CPAL-CNN is applied for $N_{AL} = 400$, which is about 7.5%, with respect to random sampling. It can also be seen that the performance

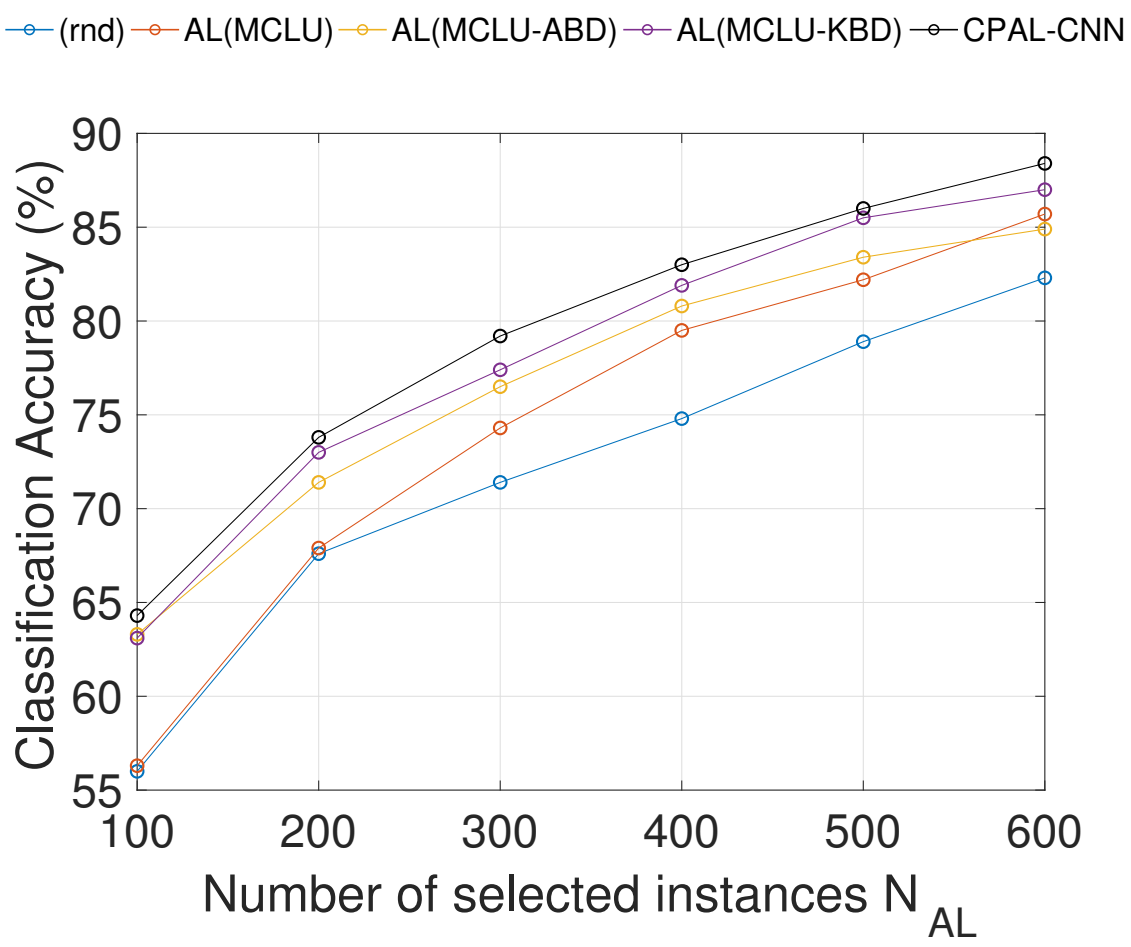


Figure 4.1: Classification accuracy (%) as a function of N_{AL} for different query functions (YaleB).

of CPAL-CNN is highest among all the considered approaches for the different values of N_{AL}. For instance, for $N_{AL} = 400$, the classification accuracy of AL(MCLU), AL(MCLU-ABD), and AL(MCLU-KBD) is 70.8%, 70.7%, and 72.2%, respectively, whereas that of CPAL-CNN is 73.9%.

Similar results are obtained for Caltech101, as shown in Fig. 4.3. The largest performance improvement is obtained when CPAL-CNN is applied for $N_{AL} = 400$,

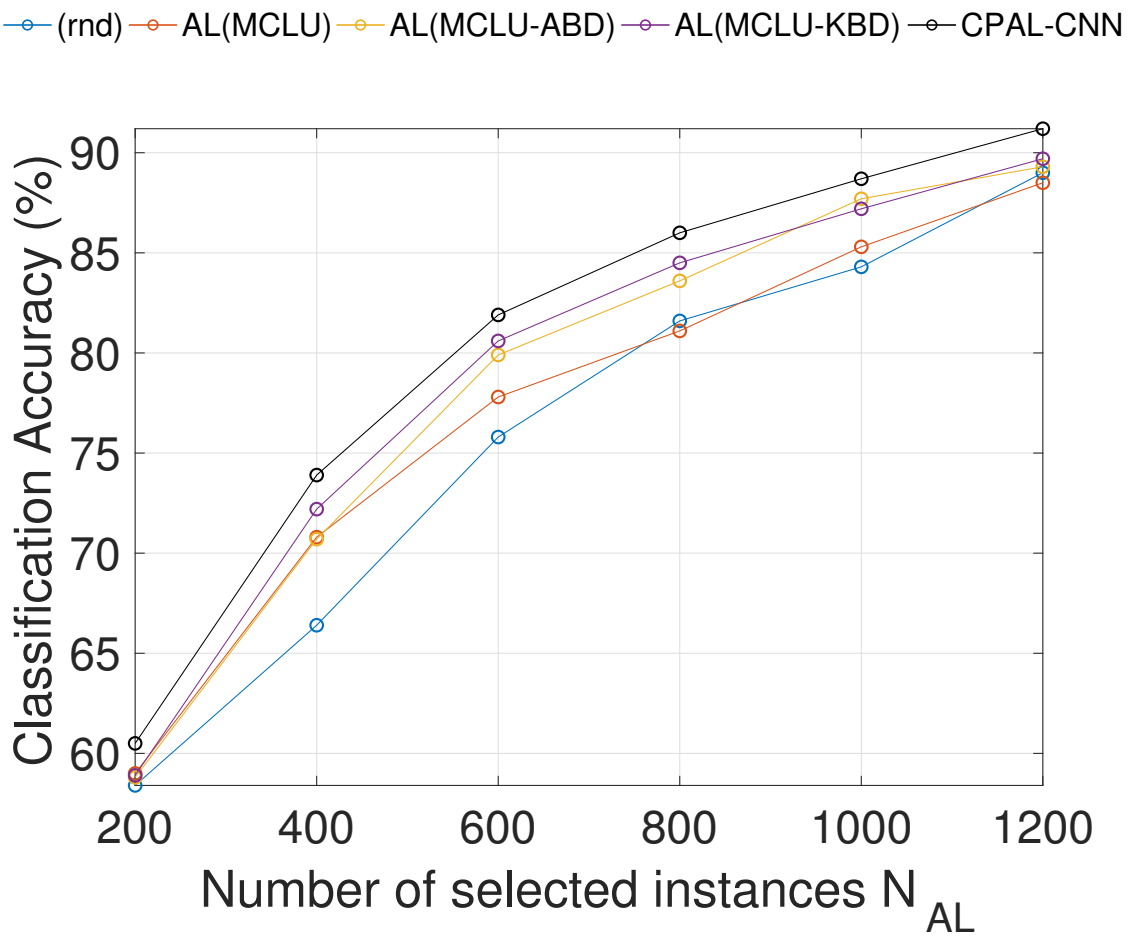


Figure 4.2: Classification accuracy (%) as a function of N_{AL} for different query functions (AR).

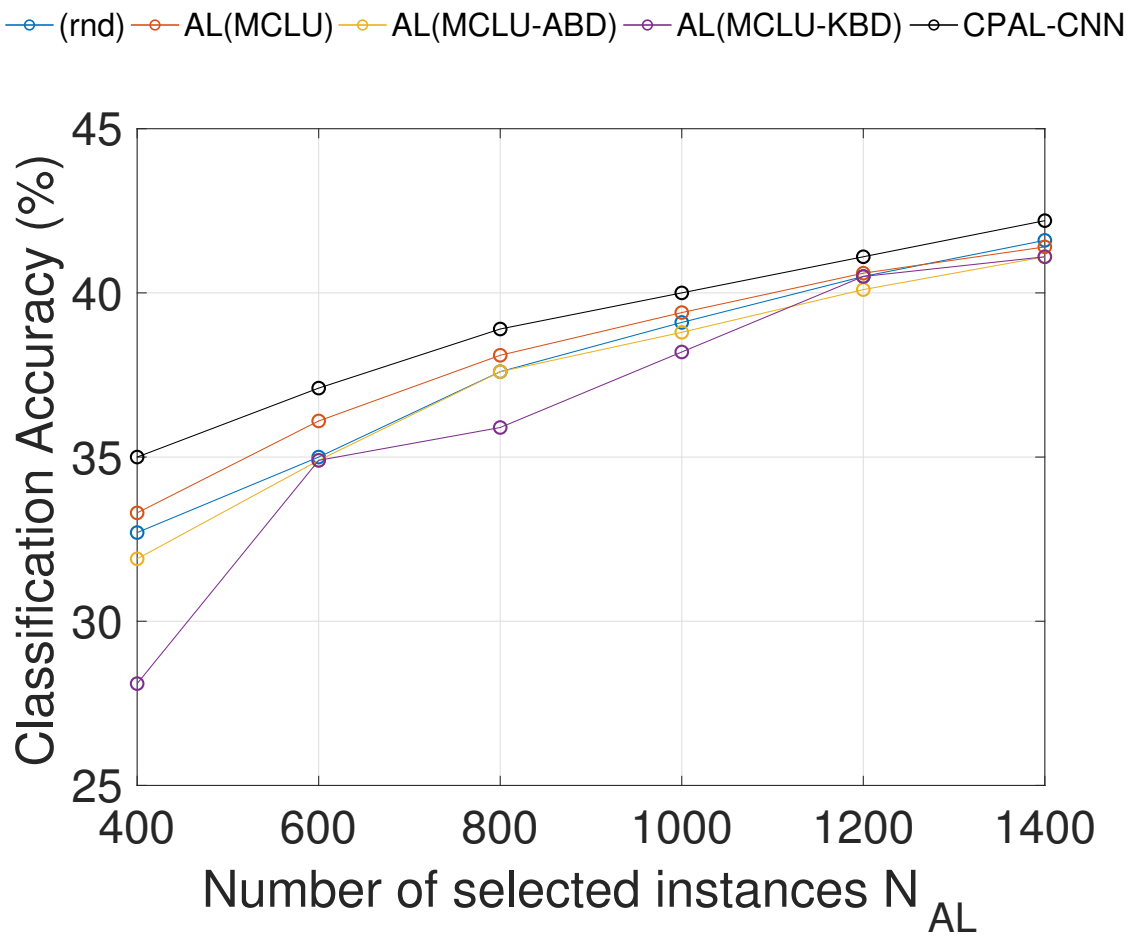


Figure 4.3: Classification accuracy (%) as a function of N_{AL} for different query functions (Caltech101).

Table 4.4: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}.

Query function	YaleB			AR			Caltech101		
	No. instances N_{AL}			No. instances N_{AL}			No. instances N_{AL}		
	300	400	500	400	600	800	400	600	800
(rnd)	71.4	74.8	78.9	66.4	75.8	81.6	32.7	35.0	37.6
AL(MCLU)	74.3	79.5	82.2	70.8	77.8	81.1	33.3	36.1	38.1
AL(MCLU-ABD)	76.5	80.8	83.4	70.7	79.9	83.6	31.9	34.9	37.6
AL(MCLU-KBD)	77.4	81.9	85.5	72.2	80.6	84.5	28.1	34.9	35.9
CPAL-CNN	**79.2**	**83.0**	**86.0**	**73.9**	**81.9**	**86.0**	**35.0**	**37.1**	**38.9**

which is about 2.3%, with respect to random sampling. As in the previous experiments, the performance of CPAL-CNN is the best among all the considered approaches. For instance, for $N_{AL} = 400$, the classification accuracy of AL(MCLU), AL(MCLU-ABD), and AL(MCLU-KBD) is 33.3%, 31.9%, and 28.1%, respectively, whereas that of CPAL-CNN is 35.0%.

Figure 4.4 shows the classification accuracy of CPAL-CNN as a function of the parameters α and β for the Extended YaleB, AR, and Caltech101 databases. It is observed that the best performance is obtained for a combination of uncertainty, diversity, and information density, *i.e.*, $\alpha, \beta \in (0, 1)$, for all the considered databases. The best performance is obtained when $\alpha = 0.4$, and $\beta = 0.3$, for the Extended YaleB database (64.3%), $\alpha = 0.6$, and $\beta = 0.2$, for the AR database (73.9%), and $\alpha = 0.2$, and $\beta = 0.4$, for the Caltech101 database (35.0%).

4.4.3 Results: Dimensionality Reduction for DML and Computational Load

DML is performed in a reduced space, obtained through PCA, to lower the computational load. The dimensionality of the vectorized images is reduced until 95% of the variance is explained. The size of the vectorized images in the Extended YaleB, AR, and Caltech101 databases is reduced from 1024×1, 2500×1, and 3072×1 to 59×1, 76×1, and 214×1, respectively. Table 4.5 shows the execution time (total

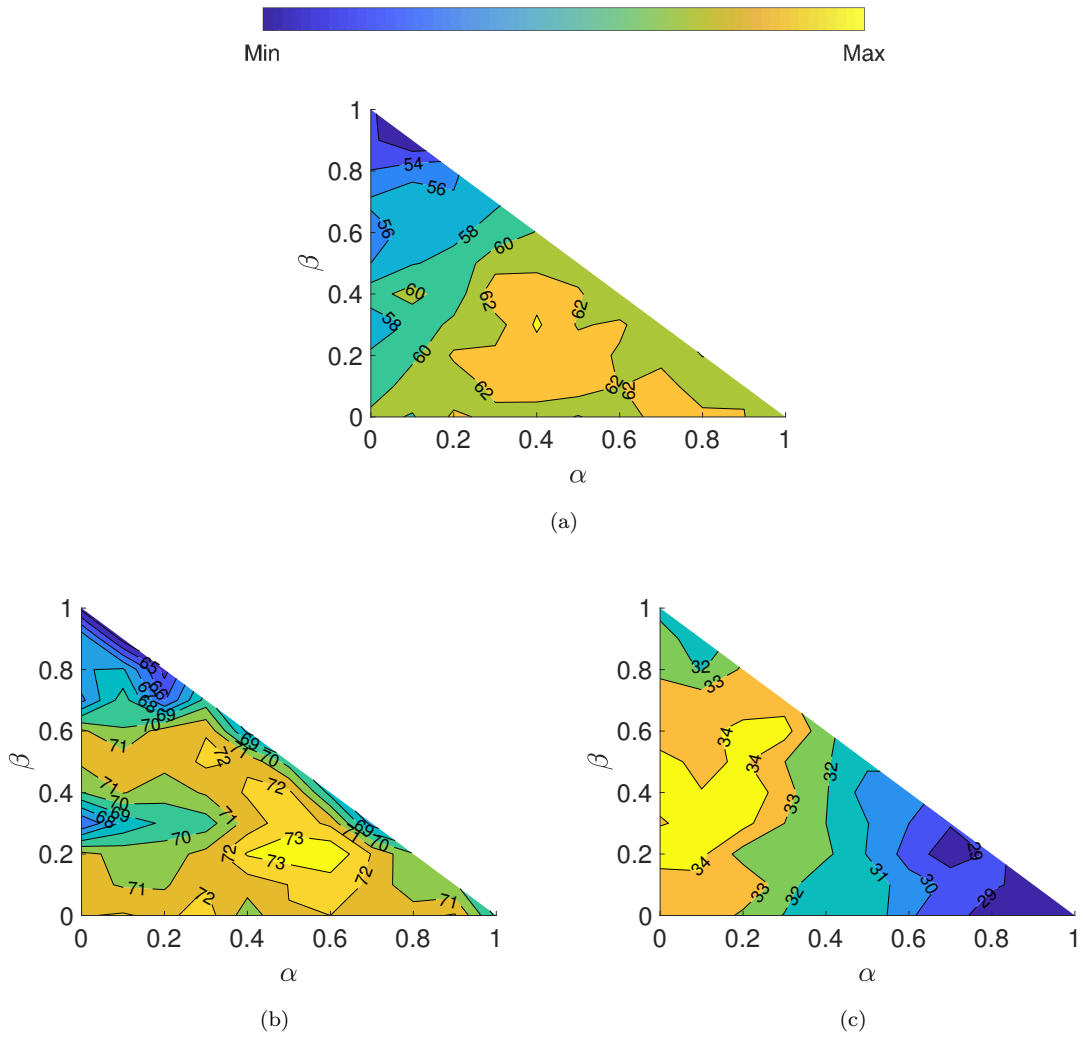


Figure 4.4: Classification accuracy (%) of CPAL-CNN as a function of α and β, (a) YaleB ($N_{AL} = 100$), (b) AR ($N_{AL} = 400$), (c) Caltech101 ($N_{AL} = 400$).

and average per iteration), speed-up (obtained through PCA), and number of iterations for convergence of LMNN for the aforementioned databases. It is observed that LMNN converges significantly faster when PCA is used. The speed increase of LMNN (total/avg. iter) obtained through PCA for the Extended YaleB, AR, and Caltech101 databases is 4.5x/2.1x, 35.7x/11.4x, and 10.9x/6.7x, respectively. In addition, the results in Table 4.5 show that the number of iterations for convergence of LMNN is decreased when PCA is employed.

Table 4.5: Execution time, speed-up, and number of iterations for convergence of LMNN.

Database		Execution time (s)		Iterations
		total	avg. iter.	
YaleB	PCA	99.2	0.7	146
	original	446.5	1.5	198
	speed-up	4.5x	2.1x	-
AR	PCA	75.3	0.8	97
	original	2691.9	9.1	296
	speed-up	35.7x	11.4x	-
Caltech101	PCA	527.8	2.9	181
	original	5792.2	19.5	297
	speed-up	10.9x	6.7x	-

4.4.4 Results: Quality of CPAL-CNN confidence values

In this section, the quality of the confidence values produced by CPAL-CNN is compared with that of the confidence values obtained through the hinge and margin nonconformity measures. Experiments are performed on the Extended YaleB, AR, and Caltech101 databases. Different significance levels, $\epsilon \in [0, 1]$, are used yielding different prediction sets Ψ^{ϵ}_{n+j}, for test instances $\mathbf{x}_{n+j}$. Notice that the hinge and margin nonconformity measures are particular cases of the CPAL-CNN nonconformity measure described by equation (4.1), when $\gamma = 1.0$ and $\gamma = 0.5$, respectively. The quality of the CPAL-CNN confidence values is demonstrated using three metrics: [12] [69]:

- *ValE*: The percentage of errors measured as the number of times the correct label for instances $\mathbf{x}_{n+j}$ is not in Ψ^{ϵ}_{n+j}, for a given ϵ, divided by the total number of test instances [12] (ValE $\approx \epsilon$, according to the validity property)
- *SinP*: The proportion of all predictions that are singletons, *i.e.,* instances $\mathbf{x}_{n+j}$ that produce $|\Psi^{\epsilon}_{n+j}| = 1$, for a given $\epsilon \in [0, 1]$. The motivation for this metric is that singleton predictions are the most informative [69] (high SinP is preferable).
- *AvgC*: The average number of class labels in the prediction sets Ψ^{ϵ}_{n+j}, as a percentage of the total number of classes, *i.e.*, a direct measure of how good the model is at rejecting class labels (low AvgC is preferable)

For the following experiments, different significance levels, ϵ, are used yielding different sets Ψ^{ϵ}_{n+j} for the test instances $\mathbf{x}_{n+j}$. The CPAL-CNN nonconformity measure

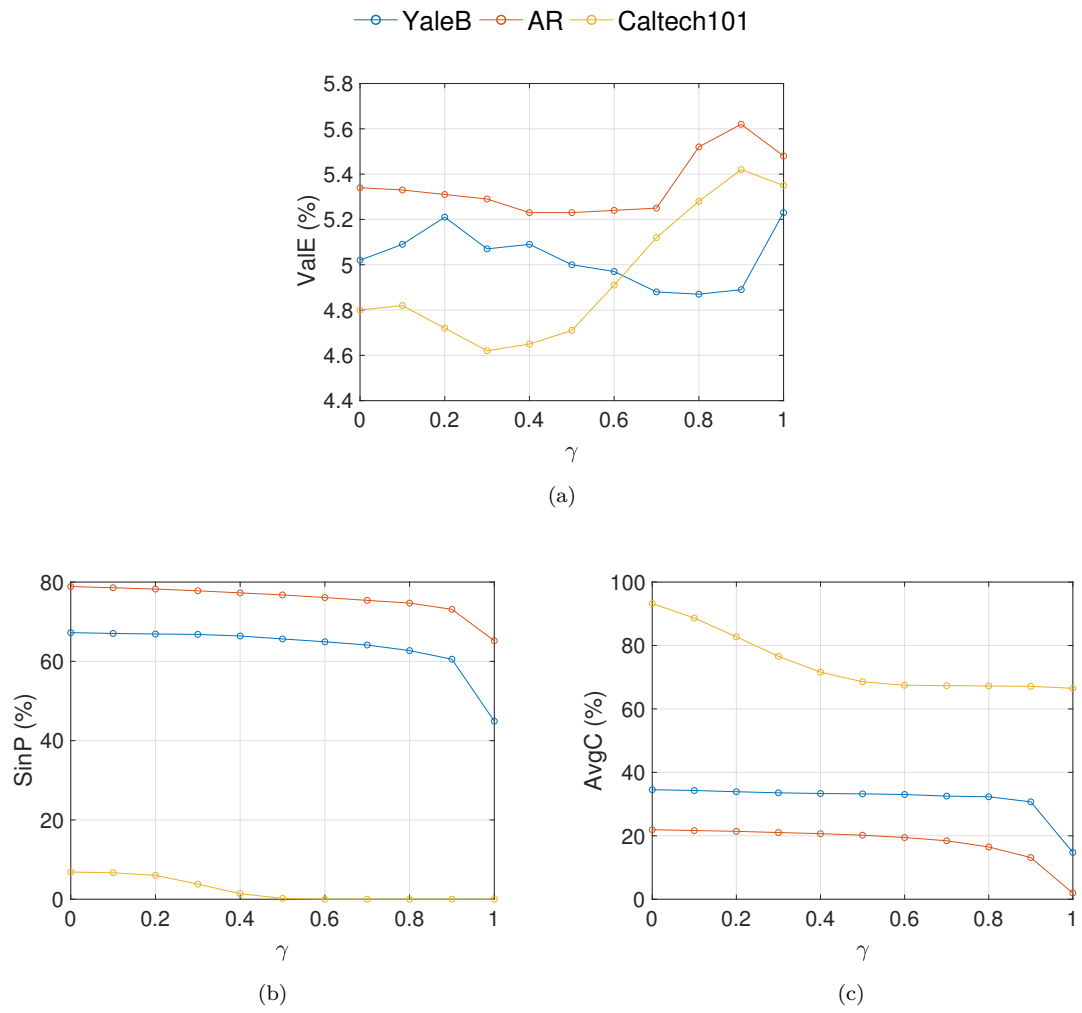


Figure 4.5: Performance of the proposed nonconformity measure for $\epsilon = 0.05$ as a function of the parameter $\gamma \in [0, 1]$ using different metrics: (a) ValE, (b) SinP, (c) AvgC.

given by (4.1) is evaluated along with the hinge and margin nonconformity measures for comparison. Notice that the hinge and margin nonconformity measures are particular cases of (4.1) when $\gamma = 1.0$ and $\gamma = 0.5$, respectively.

Figure 4.5 shows the performance of the proposed nonconformity measure, as a function of the parameter $\gamma \in [0, 1]$, using the three aforementioned metrics. It is observed in Fig. 4.5(a) that the validity property is satisfied for all the considered databases and values of γ (ValE $\approx \epsilon$). This demonstrates the usefulness of the

Table 4.6: Performance of hinge, margin, and CPAL-CNN nonconformity measures.

Database	Confidence (%)	Performance (%)								
		Hinge			Margin			CPAL-CNN		
		ValE	SinP	AvgC	ValE	SinP	AvgC	ValE	SinP	AvgC
YaleB	98	**2.3**	33.7	**22.2**	2.6	55.4	31.4	**2.3**	**57.1**	**22.2**
	95	5.2	44.9	**14.8**	**5.0**	65.6	33.2	**5.0**	**67.2**	**14.8**
	90	**9.9**	58.1	**8.1**	10.4	76.4	21.1	**9.9**	**79.6**	**8.1**
AR	98	3.0	47.0	**3.6**	2.7	66.1	31.4	**2.5**	**67.8**	**3.6**
	95	6.1	65.2	**1.9**	**5.2**	76.7	20.2	**5.2**	**78.9**	**1.9**
	90	10.6	81.6	**1.3**	10.4	88.1	7.4	**10.3**	**90.3**	**1.3**
Cal101	98	1.7	0	**83.1**	1.6	0.1	84.5	**1.9**	**3.6**	**83.1**
	95	5.3	0	**66.5**	4.7	0.2	68.4	**4.9**	**6.9**	**66.5**
	90	8.4	0	**57.9**	7.7	0.2	58.7	**8.8**	**10.9**	**57.9**

CPAL-CNN confidence values. Fig. 4.5(b) shows the behavior of singleton predictions as a function of γ (SinP). It is observed that as the value of γ increases, the percentage of singleton predictions decreases, meaning that lower values of γ produce higher number of singleton predictions. The highest number of singletons is obtained for $\gamma = 0$, with SinP = 67.2%, SinP = 78.9%, and SinP = 6.9% for the Extended YaleB, AR, and Caltech101 databases, respectively. The average number of class labels in the prediction sets, as a percentage of the total number of classes (AvgC), is shown in Fig. 4.5(c), for different values of γ. Notice that AvgC decreases as the value of γ increases, *i.e.*, higher values of γ produce more discriminative predictions sets Ψ^{ϵ}_{n+j}. The smallest prediction sets Ψ^{ϵ}_{n+j} are obtained for $\gamma = 1$, with AvgC = 14.8%, AvgC = 1.9%, and AvgC = 66.5% for the Extended YaleB, AR, and Caltech101 databases, respectively. Based on the results presented in Fig. 4.5, it is observed that the parameter γ can be used to select between discriminativeness (high values of γ, low AvgC) and high number of singleton predictions (low values of γ, high SinP).

The performance results of the hinge, margin, and CPAL-CNN nonconformity measures are summarized in Table 4.6. For CPAL-CNN, the best results are shown (from those obtained using different values of γ). The results in Table 4.6 show that

CPAL-CNN achieves similar or better performance than that of the hinge and margin nonconformity measures, for all the considered performance metrics.

4.5 Chapter Conclusion

A conformal prediction based active learning algorithm for convolutional neural networks, referred to as CPAL-CNN, is proposed in this chapter. CPAL-CNN, uses a novel nonconformity measure that produces reliable confidence values. Furthermore, CPAL-CNN selects instances from an unlabeled pool based on the evaluation of three criteria: uncertainty, diversity, and representativeness. Different from previous work on active learning, the proposed query function employs DML to obtain similarity measures that adapt to the statistics of the database being used. DML is performed in a reduced space, obtained through PCA, thereby lowering the computational load.

Experiments conducted on two face databases, the Extended YaleB database and the AR database, and one object recognition database, Caltech101, demonstrate the improved performance obtained through CPAL-CNN. Moreover, it is shown that the proposed query function for CPAL-CNN outperforms previous work on active learning, increasing classification performance across different CNN architectures and databases.

In addition to performance enhancement, CPAL-CNN produces reliable confidence values that are used to predict class labels with guaranteed error rate. The quality of the CPAL-CNN confidence values is demonstrated experimentally using three different metrics: the evaluation of the validity property, the percentage of singleton predictions, and the average number of class labels in the prediction sets. The results show that CPAL-CNN achieves similar or better performance than that obtained using previously proposed techniques.

In the following chapter, a constrained linear regression model is considered to determine the relevance of instances in the unlabeled pool. Moreover, experiments conducted on synthetic databases are performed to give a greater insight into the effect of uncertainty, diversity, and representativeness on the selection of instances.

Chapter 5

CONFORMAL PREDICTION BASED ACTIVE LEARNING BY LINEAR REGRESSION OPTIMIZATION

We propose a conformal prediction based active learning algorithm that obtains a measure of the relevance of an instance through the solution of a constrained linear regression model. This approach is referred to as CPAL-LR. The proposed technique considers uncertainty, diversity, and representativeness as the selection criteria. In the remainder of this section, the proposed query function is introduced, and the CPAL-LR algorithm is described.

5.1 CPAL-LR Query Function

The proposed query function determines the relevance of unlabeled instances through the solution of a constrained linear model, incorporating uncertainty, diversity, and representativeness in the optimization problem. Define $U = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ as the unlabeled pool. Let $\mathbf{Q} \in \mathbb{R}^{L \times L}$ be a kernel distance matrix, containing the distances between each one of the elements in the unlabeled pool. The entries $q_{ij} \in [0, 1]$ in matrix $\mathbf{Q}$ are computed as:

$$\mathcal{K}_\eta(\mathbf{x}_i, \mathbf{x}_j) = q_{ij} = \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}{\eta}\right). \tag{5.1}$$

Let $\mathbf{y} \in \mathbb{R}^L$ be a vector consisting of elements y_i, containing the value of uncertainty associated with instances $\mathbf{x}_i \in U$ $(i = 1, \ldots, L)$, calculated according to equation (2.3). Let $\mathbf{D} \in \mathbb{R}^{L \times L}$ be a positive diagonal matrix, whose diagonal elements $d_i \in [1, 0]$ provide a measure of the representativeness (information density) of instances $\mathbf{x}_i$. The value d_i decreases when instance $\mathbf{x}_i$ is located in a densely populated region,

otherwise the value d_i increases. The proposed approach obtains a vector $\hat{\mathbf{w}} \in \mathbb{R}^L$, consisting of elements $\hat{w}_i$, containing the relevance values associated with instances $\mathbf{x}_i$ by solving the following optimization problem:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \ \|\mathbf{Q}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{D}\mathbf{w}\|_2^2 \qquad (5.2)$$
$$s.t.\ \mathbf{0} \leq \mathbf{w} \leq \mathbf{1},$$

which is a generalized ridge regression problem, penalized by the diagonal matrix $\mathbf{D}$.

Expanding the first term in equation (5.2) we have

$$(\mathbf{Q}\mathbf{w} - \mathbf{y}) = \underbrace{\begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1L} \\ q_{21} & q_{22} & \cdots & q_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ q_{L1} & q_{L2} & \cdots & q_{LL} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix}}_{\text{diversity}} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_L \end{bmatrix}.$$

Notice that the values w_j are weighed by the terms q_{ij}. The weights q_{ij} increase when instances $\mathbf{x}_i$ and $\mathbf{x}_j$ are close to each other ($q_{ij} = 1$, for $i = j$). Since the solution $0 \leq \tilde{w}_j \leq 1$, the instances whose uncertainty is high, and are also different from each other, receive a low penalty. Conversely, the instances that are close to each other, *i.e.*, they are not diverse, receive a higher penalty. Therefore, the term $\|\mathbf{Q}\mathbf{w} - \mathbf{y}\|_2^2$ accounts for diversity, and the parameter η in equation (5.1) provides a tradeoff between uncertainty and diversity.

Expanding the second term in (5.2) we obtain

$$\mathbf{Dw} = \underbrace{\begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & d_N \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix}}_{\text{representativeness}}.$$

The elements d_i penalize the solution $\hat{\mathbf{w}}_i$ depending on the representativeness of instance $\mathbf{x}_i$. When $\mathbf{x}_i$ is located in a densely populated region, the value d_i decreases (representative, low penalty). Conversely, when x_i is located in a sparsely populated region, d_i increases (not representative, high penalty). Therefore, the parameter λ controls the penalty associated with representativeness, which is used to filter possible outliers.

The expression in (5.2) can be rewritten as:

$$\begin{aligned} \hat{\mathbf{w}} &= \arg\min_{\mathbf{w}} \ \left\| \tilde{\mathbf{Q}}\mathbf{w} - \tilde{\mathbf{y}} \right\|_2^2 \\ &= \arg\min_{\mathbf{w}} \ \mathbf{w}^T \tilde{\mathbf{Q}}^T \tilde{\mathbf{Q}}\mathbf{w} - \mathbf{w}^T \tilde{\mathbf{Q}}^T \tilde{\mathbf{y}} \\ &s.t.\ \mathbf{0} \leq \mathbf{w} \leq \mathbf{1}, \end{aligned} \tag{5.3}$$

where $\tilde{\mathbf{Q}} = \left[\mathbf{Q}\sqrt{\lambda}\mathbf{D}\right]^T \in \mathbb{R}^{2L \times L}$, and $\tilde{\mathbf{y}} = [\mathbf{y}\ \mathbf{0}]^T \in \mathbb{R}^{2L}$. Notice that the expression in (5.3) is a quadratic programming (QP) optimization problem.

After the optimization problem in (5.3) is solved, the number of desired instances, N_{AL}, associated with the highest relevance values w_j $(j = 1, \dots, L)$ are selected.

5.2 Incorporating Representativeness

The second term in equation (5.2) penalizes the solution $\hat{\mathbf{w}}$ through the weights d_i in matrix $\mathbf{D}$. CPAL-LR computes the weights d_i, associated with instances $\mathbf{x}_i$ in the unlabeled pool, using the distance between $\mathbf{x}_i$ and its k-nearest neighbors, denoted as $\mathbf{z}_i^{(j)}$ [61], for $j = 1, \ldots, k$. Define the value $\hat{d}_i$, associated with instance $\mathbf{x}_i$, as:

$$\hat{d}_i = \sum_{n=1}^{k} \left\| \mathbf{x}_i - \mathbf{z}_i^{(n)} \right\|_2^2. \tag{5.4}$$

Notice that the value $\hat{d}_i$ will be low if instance $\mathbf{x}_i$ is close to its k-nearest neighbors (densely populated region, low penalty). Conversely, the value $\hat{d}_i$ will be high if instance $\mathbf{x}_i$ is far from its k-nearest neighbors (sparsely populated region, high penalty). Define $d_{max} = \max\{d_i\}$. CPAL-LR computes the values $\hat{d}_i$ for all instances $\mathbf{x}_i$ in the unlabeled pool ($i = 1, \ldots, L$) as:

$$d_i = \hat{d}_i / d_{max}. \tag{5.5}$$

5.3 CPAL-LR Nonconformity Measure

We use the nonconformity measure described by equation (4.1). Notice that, regardless of the type of classifier, the nonconformity scores are normalized through the computation of p-values, which are then used to measure uncertainty, according to (2.3). For instance, the j-th output of a linear classifier to input $\mathbf{x}$ can be defined as $o_j = \mathbf{w}_j \mathbf{x} + b_j \in \mathbb{R}$, whereas the j-th output of a CNN is obtained through its forward propagation function, and it is taken directly from its last layer (usually a softmax). In both cases, the value of uncertainty $I(\cdot)$ computed within the CP framework is normalized in the range $[0, 1]$, and can be readily used for active learning without further scaling.

5.4 CPAL-LR Algorithm

We propose an active learning algorithm within the CP framework. First, we split the training set, $T_{train} = \{z_1, ..., z_n\}$, into the proper training set, $T_{prop} = \{z_1, ..., z_\ell\}$, and the calibration set, $T_{cal} = \{z_{\ell+1}, ..., z_{\ell+r}\}$, where $n = \ell + r$, as described in Section 2. Then, the classification rule, C_{prop}, is obtained through the underlying algorithm employing T_{prop}.

The nonconformity scores of the instances in calibration set, T_{cal}, and the unlabeled pool, U, are computed using equation (4.1) and C_{prop}. The nonconformity scores are used to measure the p-values and the uncertainty of instances in the unlabeled pool, according to equation (2.1) and (2.3), respectively.

Matrix $\mathbf{Q}$ is computed using the Gaussian kernel distance as described by (5.1), and matrix $\mathbf{D}$ is computed using the k-nearest neighbors approach, according to equations (5.4) and (5.5). Then, the quadratic optimization problem described by equation (5.3) is solved to obtain the relevance $\hat{\mathbf{w}}$ of the instances in the unlabeled pool. The N_{AL} instances $\mathbf{x}_i$ whose relevance is highest are selected.

CPAL-LR returns the training set $T_{AL} = T_{prop} \cup T_d$, where T_d is the set of pairs containing the N_{AL} instances from U, with their corresponding class labels, whose associated relevance $\hat{\mathbf{w}}$ is the highest after solving the optimization problem in (5.3). The proposed approach is summarized in Algorithm 3.

5.5 CPAL-LR as a Conformal Predictor

The proposed nonconformity measure, described by equation (4.1), can be used to produce confidence values associated with new predictions, during the testing phase. After training the underlying algorithm and obtaining a classification rule, denoted as C_{train}, the nonconformity scores $\alpha_{n+j}^{(\mathcal{H}_q)}$ and p-values $p(\alpha_{n+j}^{(\mathcal{H}_q)})$, associated with a new instance $\mathbf{x}_{n+j}$, are computed according to equations (4.1) and (2.1), respectively. Then, for a given significance level $\epsilon \in [0, 1]$, we form a set of labels $\Psi_{n+j}^{\epsilon} = \{i : p(\alpha_{n+j}^{(\mathcal{H}_i)}) > \epsilon\}$ containing the correct class label for $\mathbf{x}_{n+j}$ with probability $(1 - \epsilon)$, according to the validity property. CPAL-LR as a conformal predictor is described in Algorithm 4.

Algorithm 3 CPAL-LR

1: **Input:** Proper training set $T_{prop} = \{z_1, ..., z_\ell\}$, calibration set $T_{cal} = \{z_{\ell+1}, ..., z_{\ell+r}\}$, unlabeled pool $U = \{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+v}\}$, classification rule C_{prop}, number of desired instances N_{AL}, and number of class labels M

2: Compute matrix $\mathbf{Q}$ using equation (5.1)

3: Compute the weights d_i, using equations (2.8), and (2.9), for all instances in the unlabeled pool U to form $\mathbf{D}$

4: Use Equation (4.1) and the classification rule C_{prop} to calculate:

- The nonconformity scores $\{\alpha_{\ell+1}, \ldots, \alpha_{\ell+r}\}$ corresponding to the instances in the calibration set.
- The nonconformity scores $\{\alpha_{n+1}^{\mathcal{H}_i}, \ldots, \alpha_{n+v}^{\mathcal{H}_i}\}$ corresponding to the instances in the unlabeled pool, where $i = \{1, \ldots, M\}$

5: Use Equation (2.1) to calculate the p-values associated with the instances in U, and obtain their uncertainty $I(\mathbf{x}_{n+j})$ through equation (2.3), where $j \in \{1, \ldots, v\}$

6: Solve the quadratic optimization problem in (5.3) and form the set T_d containing the N_{AL} instances from U, with their corresponding class labels, whose associated relevance w is the highest

7: Construct $T_{AL} = T_{prop} \cup T_d$

8: **Output:** T_{AL}

Algorithm 4 CPAL-LR (conformal predictor)

1: **Input:** Testing instance $\mathbf{x}_{n+j}$, calibration set nonconformity scores $\{\alpha_{\ell+1}, \ldots, \alpha_{\ell+r}\}$, classification rule C_{train}, significance level ϵ, parameter γ, and number of class labels M

2: Use Equations (2.1) and (4.1), along with the classification rule C_{train}, to calculate:

- The nonconformity scores $\alpha_{n+j}^{\mathcal{H}_i}$ corresponding to the new instance $\mathbf{x}_{n+j}$, for the different null hypothesis $\mathcal{H}_i$ $(i = \{1, \ldots, M\})$
- The p-values $p(\alpha_{n+j}^{(\mathcal{H}_i)})$, associated with $\alpha_{n+j}^{\mathcal{H}_i}$

3: Construct the set $\Psi_{n+j}^{\epsilon} = \{i : p(\alpha_{n+j}^{(\mathcal{H}_i)}) > \epsilon\}$

4: **Output:** Ψ_{n+j}^{ϵ}

5.6 Experimental Results

The focus of CPAL-LR is twofold: 1) to improve the performance of pattern classification algorithms through active learning; and 2) to produce reliable confidence values. Therefore, our goal is to evaluate CPAL-LR based on the improvement achieved in classification performance and the quality of the produced confidence values. This section is organized as follows. First, we present simulation results obtained on a synthetic database to provide a greater insight into the proposed query function and show

its effectiveness. Then, we evaluate the performance of CPAL-LR on face and object recognition databases, providing a comparison between the proposed technique and previous work on active learning. Last, we demonstrate the quality of the confidence values obtained through CPAL-LR.

5.6.1 Synthetic Database Experiments

Experiments are conducted on two different synthetic databases, which are described below:

The *Gaussian* database consists of four two-dimensional clusters, denoted as C_i ($i = 1 \ldots 4$). The data in C_i is randomly generated following a multivariate gaussian distribution given by $\mathcal{N} \sim (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, where $\boldsymbol{\mu}_i = (\mu_i^{(1)}, \mu_i^{(2)})$ and $\boldsymbol{\Sigma}_i = \begin{pmatrix} \sigma_i & 0 \\ 0 & \sigma_i \end{pmatrix}$ are the mean and covariance matrix of C_i, respectively. The parameters of the synthetic database are set to $\boldsymbol{\mu}_1 = (-0.3, 0.3)$, $\boldsymbol{\mu}_2 = (-0.3, -0.3)$, $\boldsymbol{\mu}_3 = (0.3, 0.3)$, $\boldsymbol{\mu}_4 = (0.3, -0.3)$ and $\sigma_1 = \sigma_2 = \sigma_3 = \sigma_4 = \sigma$ (different values of σ are used). The proper training set T_{prop} consists of 10 examples per class. The unlabeled pool U and the testing set consist of 200 images per class, each.

The *Two-moon* database consists of two semicircles (moons), denoted as S_i ($i = 1, 2$), with radii r_i, and center c_i. This database is generated by adding two-dimensional Gaussian noise to the two semicircles. We denote $\mathcal{N} \sim (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as the Gaussian mean and covariance matrix, where $\boldsymbol{\mu} = (\mu^{(1)}, \mu^{(2)})$ and $\boldsymbol{\Sigma} = \left(\begin{smallmatrix} \sigma & 0 \\ 0 & \sigma \end{smallmatrix}\right)$. The parameters of the synthetic database are set to $\boldsymbol{\mu} = (0, 0)$, $r_1 = r_2 = 0.5$, $c_1 = \{-0.3, 0\}$, and $c_2 = \{0.3, 0.25\}$. Various values of σ are used. The proper training set T_{prop} consists of 14 examples per class. The unlabeled pool U and the testing set consist of 200 images per class, each.

SVMs are employed for these experiments, using the one-vs-all (OVA) approach. Linear SVMs are used for the Gaussian database, whereas kernel SVMs (polynomial of order 3) are used for the Two-moon database. We compare the performance improvement obtained through CPAL-LR with that of the following batch active learning approaches: random sampling, *i.e.*, we take instances from the unlabeled pool at random,

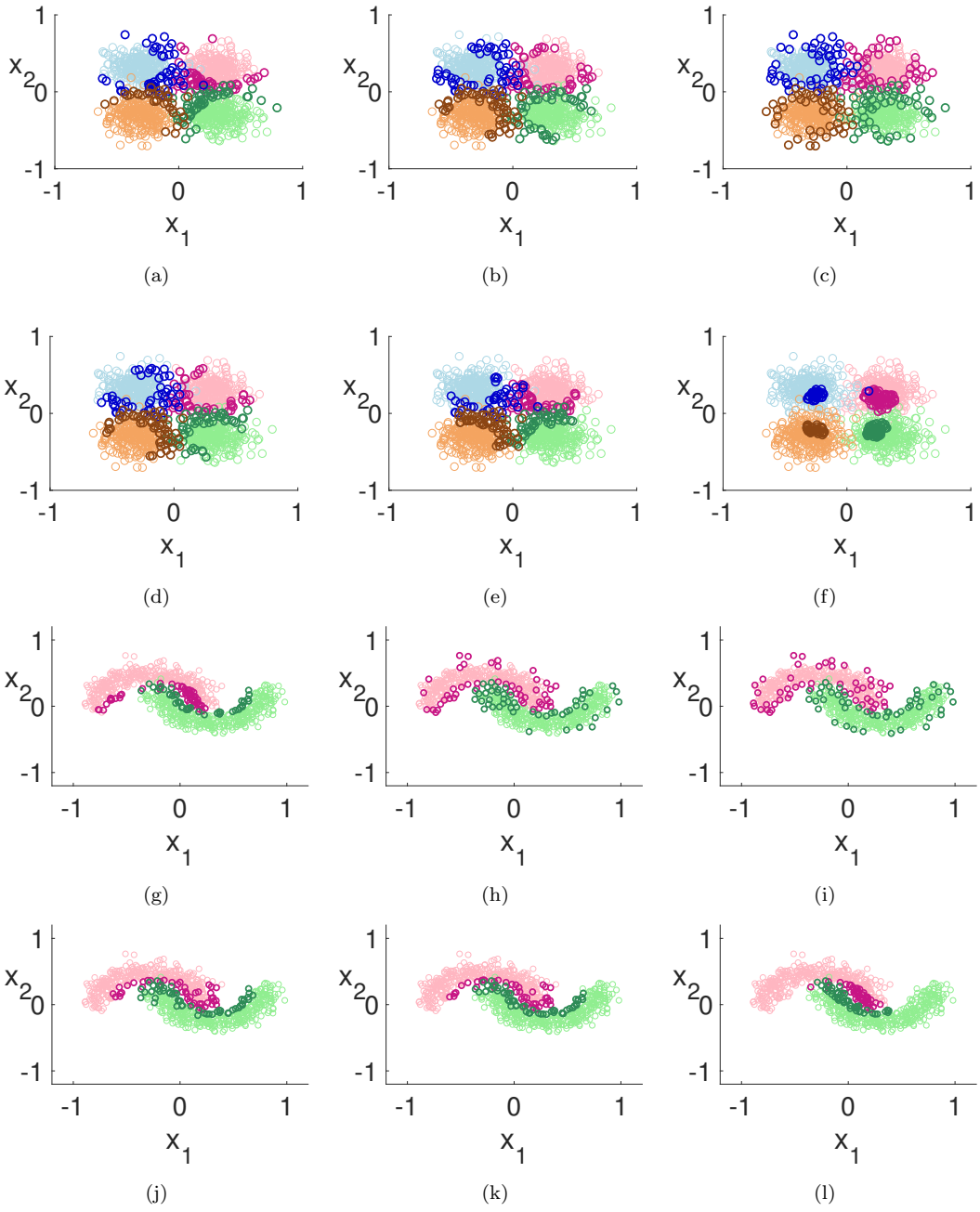


Figure 5.1: Synthetic databases and selected instances (highlighted) (Gaussian $\rightarrow$ $\sigma = 0.14$, and Two-moon $\rightarrow$ $\sigma = 0.08$) using CPAL-LR (a) ($\eta = 10^{-9}$, $\lambda = 0$), (b) ($\eta = 5.0 \times 10^{-5}$, $\lambda = 4$), (c) ($\eta = 5.0 \times 10^{-5}$, $\lambda = 0$), (d) ($\eta = 2.5 \times 10^{-5}$, $\lambda = 4$), (e) ($\eta = 5.0 \times 10^{-5}$, $\lambda = 12$), (f) ($\eta = 10^{-9}$, $\lambda = 12$), (g) ($\eta = 10^{-9}$, $\lambda = 0$), (h) ($\eta = 2.5 \times 10^{-5}$, $\lambda = 0$), (i) ($\eta = 5.0 \times 10^{-5}$, $\lambda = 0$), (j) ($\eta = 2.5 \times 10^{-5}$, $\lambda = 4$), (k) ($\eta = 5.0 \times 10^{-5}$, $\lambda = 8$), (l) ($\eta = 10^{-9}$, $\lambda = 12$).

active learning based on uncertainty [9, 12, 13], clustering [22], clustering with uncertainty [22], uncertainty and ABD [14], uncertainty and KBD [28], and active learning by sparse selection [70], which are denoted as (rnd), AL(MCLU), AL(CBD), AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), and AL(Sparse), respectively. Random sampling is used as the baseline for the experiments.

For the proposed approach, parameter optimization using exhaustive search is performed over the weights η and λ. For AL(MCLU-ABD) and AL(MCLU-KBD), the parameter ρ is optimized using the same approach. For random sampling, the training set $T_R = T_{prop} \cup T_{rnd}$ is employed, where T_{rnd} contains N_{AL} randomly selected instances from U with their corresponding class labels, and T_{prop} is the proper training set. The results for active learning are obtained using the training set $T_{AL} = T_{prop} \cup T_d$, where T_d contains N_{AL} instances selected from U using the aforementioned active learning approaches, with their corresponding class labels. Five trials are conducted to compute the classification accuracy. In each trial, the proper, calibration, training and testing sets are selected at random. For each trial, the best results are selected after parameter optimization and the average classification accuracy is presented.

Figure 5.1 shows the instances selected by the proposed technique for different parameters α and β. It is observed in Fig. 5.1(a) and (g) that when uncertainty is predominant ($\eta = 10^{-9}, \lambda = 0$), CPAL-LR selects instances that are concentrated on high uncertainty the regions, *i.e.*, the regions where clusters tend to overlap (near the decision boundaries).

Figure 5.1(f) and (l) shows the instances selected by CPAL-LR when representativeness is predominant ($\eta = 10^{-9}, \lambda = 12$). It is observed that the selected instances are located near the cluster centers for the Gaussian database (Fig. 6.1(f)), and they concentrate near coordinate (0,0) for the Two-moon database, which correspond to densely populated regions. On the other hand, when diversity is predominant ($\eta = 5.0 \times 10^{-5}, \lambda = 0$), the instances selected by CPAL-LR are located in sparsely populated regions. as shown in Fig. 6.1(c) and (i).

Table 5.1: Classification accuracy (%) for different query functions and standard deviation σ as a function of the number of selected instances N_{AL}.

Algorithm	Query function	Gaussian				Two-moon			
		$\sigma = 0.13$		$\sigma = 0.17$		$\sigma = 0.08$		$\sigma = 0.14$	
		N_{AL}		N_{AL}		N_{AL}		N_{AL}	
		12	20	12	20	12	20	12	20
SVM	(rnd)	94.3	94.9	87.1	87.7	90.9	91.6	86.3	88.4
	AL(MCLU)	93.4	95.8	87.8	88.6	91.0	91.7	87.8	88.5
	AL(CBD)	94.3	94.6	87.4	88.4	91.1	92.0	87.8	88.4
	AL(MCLU-ECBD)	94.7	95.7	88.6	89.3	92.3	93.5	88.0	89.2
	AL(MCLU-ABD)	94.7	96.1	88.7	89.5	92.4	93.2	88.1	89.7
	AL(MCLU-KBD)	95.4	95.8	89.3	89.7	92.9	93.3	88.0	88.5
	AL(Sparse)	96.0	97.0	89.9	90.2	92.7	93.8	89.8	90.1
	CPAL-LR	**97.1**	**97.4**	**90.6**	**90.9**	**93.4**	**96.4**	**90.7**	**91.0**

Figures 5.1(e) and (k) show the instances selected by CPAL-LR when uncertainty, diversity, and representativeness are considered together. The parameters for the Gaussian and Two-moon databases are ($\eta = 5.0 \times 10^{-5}, \lambda = 12$) and ($\eta = 5.0 \times 10^{-5}, \lambda = 8$), respectively. It is observed that the selected instances are located in high uncertainty regions, and the spread of the selected instances is lower. In addition, there are no instances located in sparsely populated regions.

Table 5.1 shows the classification accuracy obtained on the synthetic databases for different query functions and values σ, as a function of the number of selected instances N_{AL}. It is observed that the proposed technique outperforms the considered active learning approaches for all the values of σ and N_{AL}. For instance, when the Gaussian database is employed, for $\sigma = 0.17$ and $N_{AL} = 12$, the performance of (rnd), AL(MCLU), AL(MCLU-ABD), AL(MCLU-KBD), AL(CBD), AL(MCLU-ECBD) and AL(Sparse) is 87.1%, 87.8%, 87.4%, 88.6%, 88.7%, 89.3%, and 89.9%, respectively, whereas that of CPAL-LR is 90.6%.

To visualize the effect of the parameters η and λ on the performance of CPAL-LR we perform a second experiment. In this experiment, we conduct 100 trials. In each trial, the instances in proper, training, and testing sets are selected at random, along with those in the unlabeled pool, and the average classification accuracy is presented.

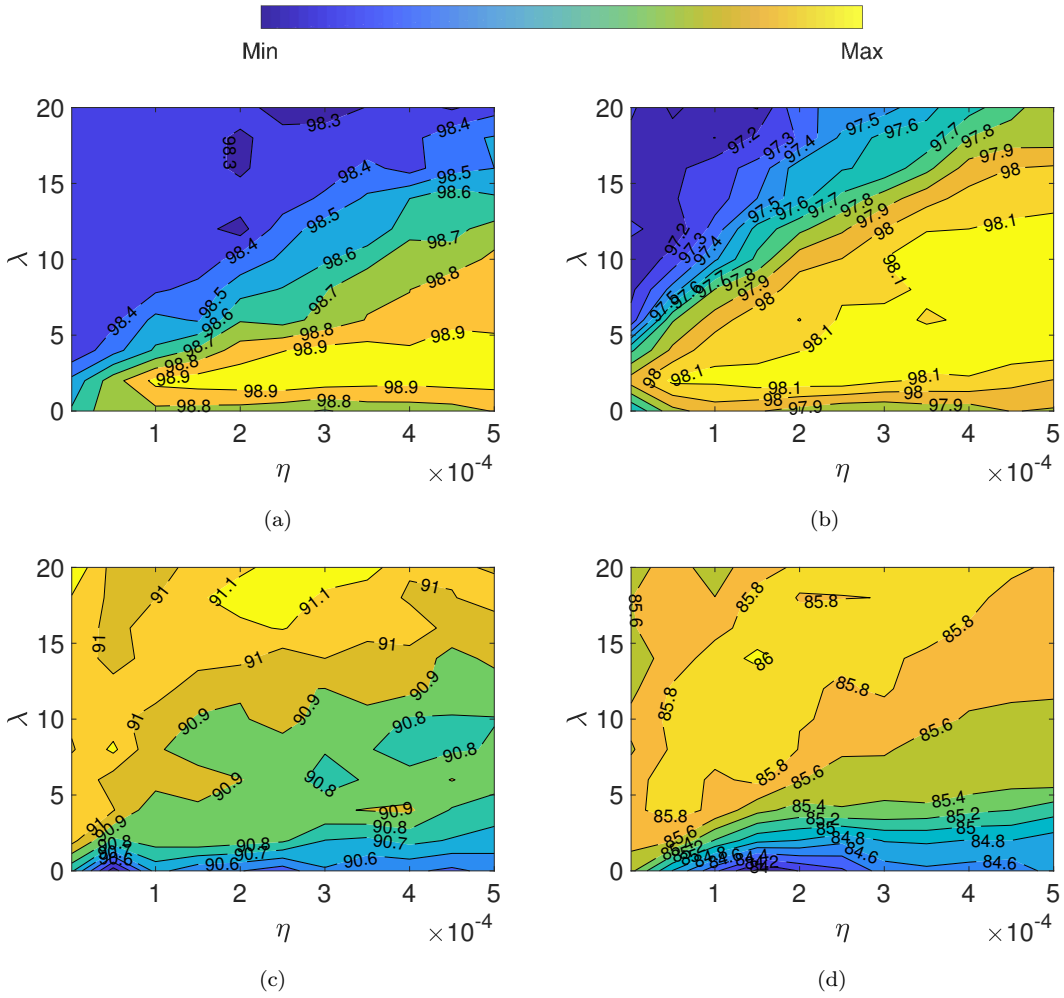


Figure 5.2: Classification accuracy (%) obtained through CPAL-LR as a function of η and λ. Gaussian: (a) $\sigma = 0.10$, (b) $\sigma = 0.12$, (c) $\sigma = 0.17$, (d) $\sigma = 0.20$.

The proper training set T_{prop} consists of 14 instances, and the number of selected instances from the unlabeled pool is $N_{AL} = 16$.

The classification accuracy for the Gaussian database as a function of η and λ for $\sigma = 0.10, 0.12, 0.17,$ and 0.20 is depicted in Fig. 5.2(a), (b), (c), and (d), respectively. The values of η and λ that produce different combinations of uncertainty, diversity, and representativeness are depicted in Fig. 5.4. It is observed that in the low variance (low noise) scenario, *i.e.*, Fig. 5.2(a) and (b), the best performance is obtained for high

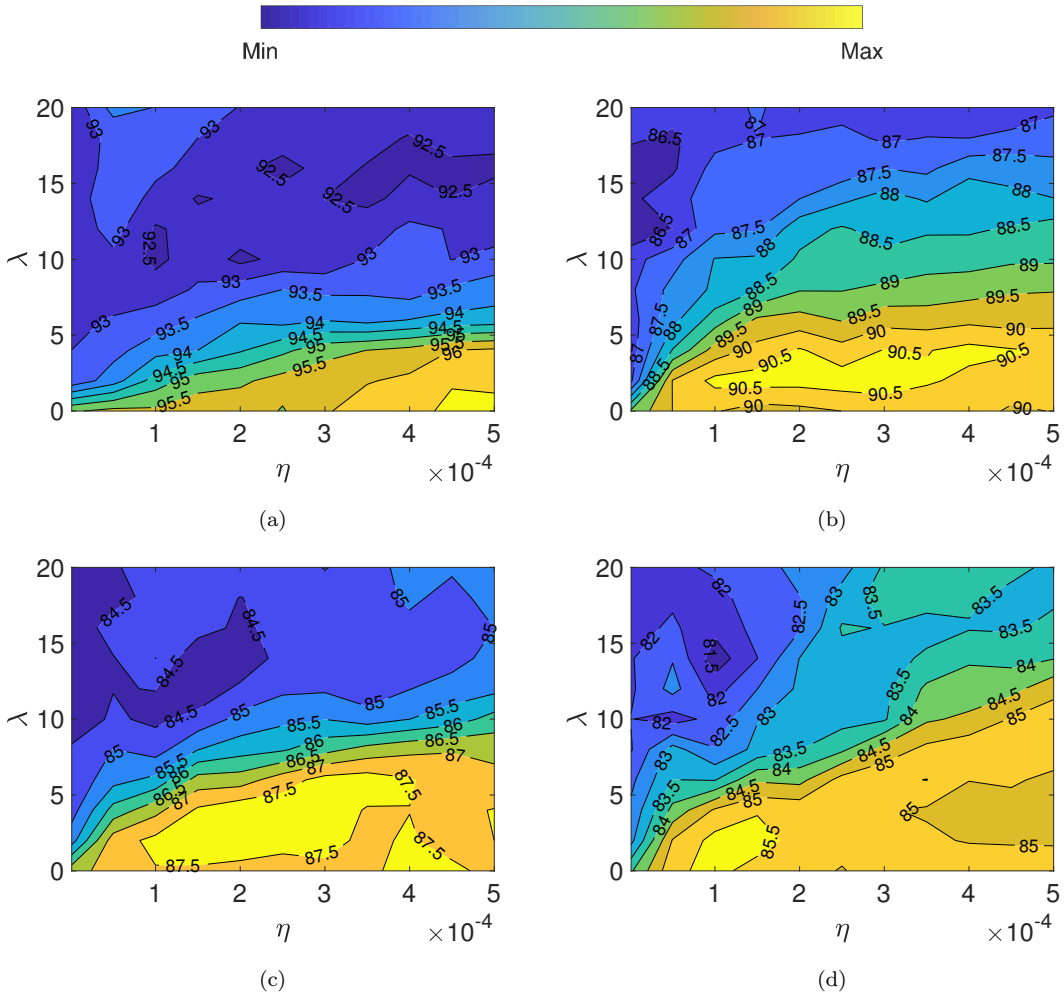


Figure 5.3: Classification accuracy (%) obtained through CPAL-LR as a function of η and λ. Two-moon: (a) $\sigma = 0.08$, (b) $\sigma = 0.13$, (c) $\sigma = 0.16$, (d) $\sigma = 0.18$.

values of η and low values of λ, which is a combination of uncertainty and diversity (towards the region $(\eta > 0, \lambda = 0)$ in Fig. 5.4). On the other hand, as the variance (noise) increases, Fig. 5.2(c) and (d), it is observed that the parameter λ becomes more relevant (towards the region $(\eta \to 0, \lambda > 0)$ in Fig. 5.4).

The classification accuracy for the Two-moon database as a function of η and λ for $\sigma = 0.08, 0.13, 0.16,$ and 0.18 is shown in Fig. 5.3(a), (b), (c), and (d), respectively. Similar to the results for the Gaussian database, it is observed that for the low noise

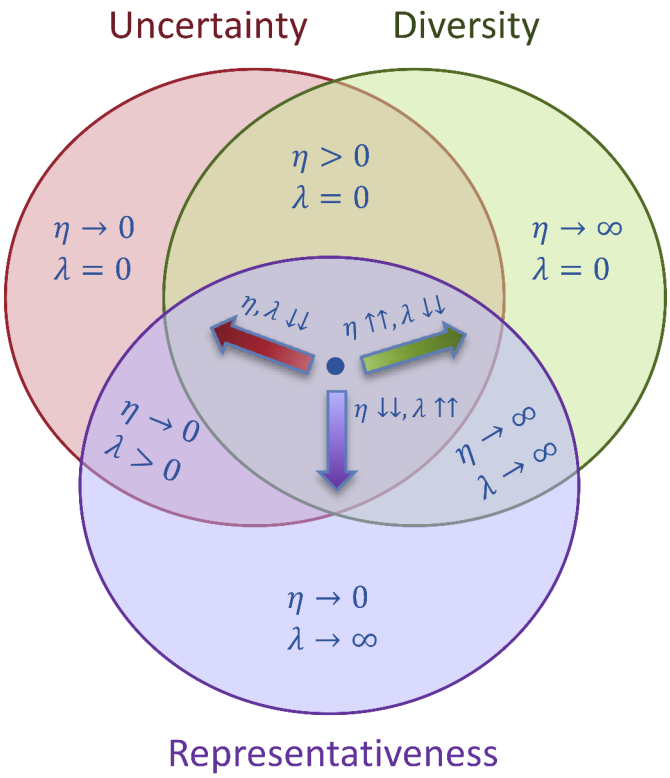


Figure 5.4: Values of η and λ that produce different combinations of uncertainty, diversity, and representativeness.

scenario, Fig. 5.3(a) and (b), high values of η produce the best results (towards the region $(\eta > 0, \lambda = 0)$ in Fig. 5.4). For the high noise case, Fig. 5.3(c) and (d), the parameter λ becomes more relevant. Different from the Gaussian database, the parameter λ does not need to be increased significantly to produce good performance in the high noise scenario for the Two-moon database.

The synthetic database experiments demonstrate that the parameters η and λ effectively control the uncertainty, diversity, and representativeness of the selected instances, providing flexibility to the proposed approach. Moreover, it is observed that CPAL-LR outperforms other existing active learning approaches for classification.

5.6.1.1 Parameter Selection Modeling for Synthetic Databases

The classification accuracy of CPAL-LR vs the parameters η and λ can be approximated by a surface defined by a fifth order polynomial in two dimensions for ease of use. The polynomial is given by the following expression:

$$\begin{aligned} f(x,y) = p_{00} + p_{10}x + p_{01}y + p_{20}x^2 + p_{11}xy + p_{02}y^2 + p_{30}x^3 + p_{21}x^2y + p_{12}xy^2 + \\ p_{03}y^3 + p_{40}x^4 + p31x^3y + p_{22}x^2y^2 + p_{13}xy^3 + p_{04}y^4 + p_{50}x^5 + p_{41}x^4y + \\ p_{32}x^3y^2 + p_{23}x^2y^3 + p_{14}xy^4 + p_{05}y^5, \end{aligned} \tag{5.6}$$

where we assign $x \leftarrow \eta \times 10^4$ (to avoid errors due to numerical precision), and $y \leftarrow \lambda$. The results of the surface-fitting exercise are shown in Fig. 5.5. It is observed that the surface defined by the fifth order polynomial closely approximates the experimental results. The coefficients of the different polynomials are summarized in Table 5.2

5.6.2 Face and Object Recognition

Experiments are conducted on two face databases, the Extended YaleB database [63] and the AR face database [64], and one object recognition database, Caltech101 [65][1]. CPAL-LR is implemented in conjunction with three different pattern classification algorithms: SVMs, sparse coding (LC-RLSDLA [51]), and CNNs. We compare the performance improvement obtained through CPAL-LR with that of the (rnd), AL(MCLU), AL(CBD), AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), and AL(Sparse). Random sampling is used as the baseline for the experiments, and parameter optimization is performed using exhaustive search, as in synthetic database experiments.

For each of the experiments in this section, five trials are conducted. In each trial, the proper, calibration, training, and testing sets are selected at random. For each trial, the best results are selected after parameter optimization and the average

[1] In this section we use a subset of the Caltech101 database including the following classes: ketch, chandelier, hawksbill, grand piano, brain, butterfly, helicopter, menorah, kangaroo, starfish, trilobite, buddha, ewer, sunflower, scorpion, revolver, laptop, ibis, llama, umbrella, crab, crayfish, cougar face, dragonfly, ferry, flamingo, and lotus.

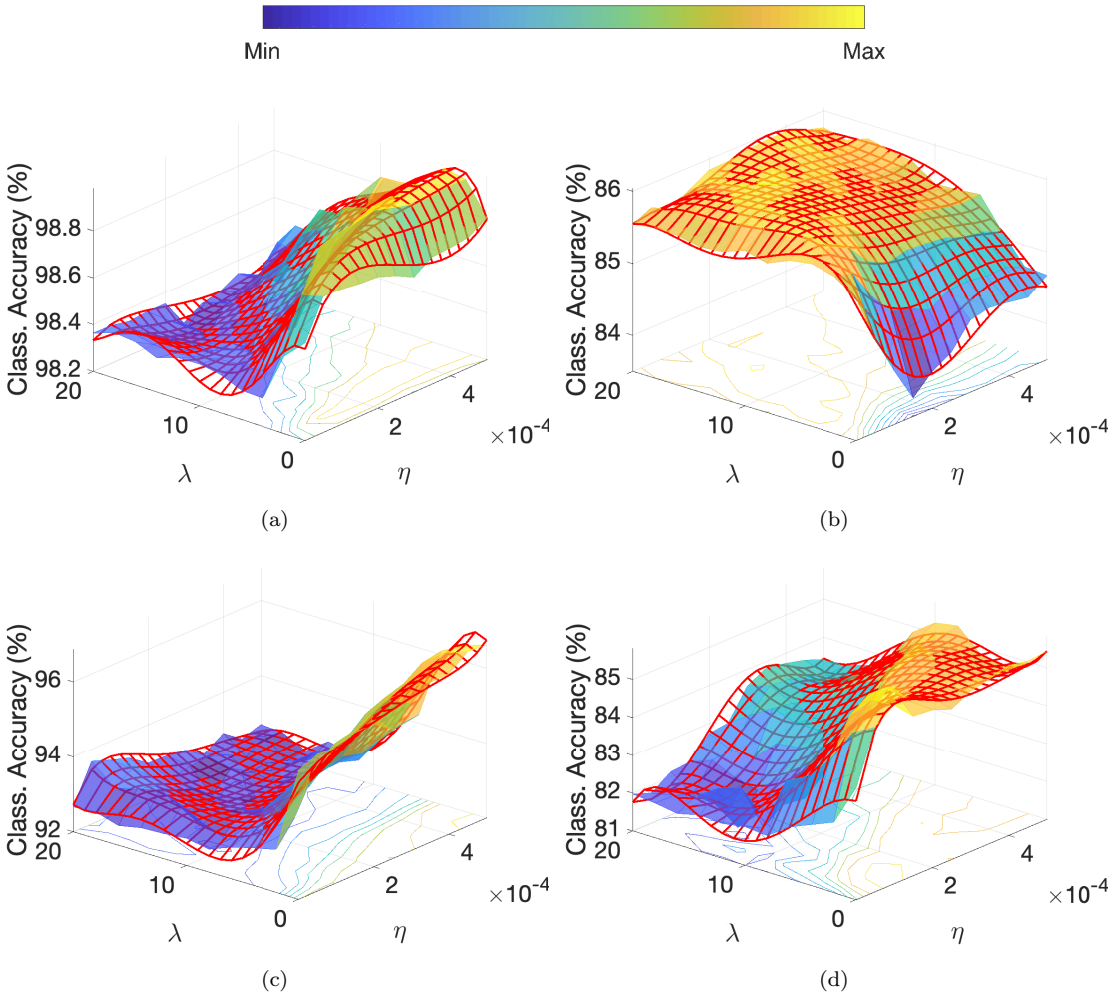


Figure 5.5: Characterization of the surface defined by performance vs parameters η and λ using a fifth order polynomial in two dimensions (fitted surface shown in red). Gaussian: (a) $\sigma = 0.10$, (b) $\sigma = 0.20$, Two-moon: (c) $\sigma = 0.08$, (d) $\sigma = 0.18$.

classification accuracy is presented. The calibration set consists of 199 instances for all the experiments, which results in a resolution of 0.5% in the confidence values calculated, according to equation (2.1). The parameter γ is set to 0.5 in the nonconformity measures given by equation (4.1). For the Extended YaleB database, the proper training set T_{prop} and U consist of eight and 24 images per class, respectively. For SVMs and LC-RLSDLA, the feature descriptors are randomfaces of size $N = 504$. The dictionary

Table 5.2: Polynomial coefficients for the synthetic databases.

Coefficients	Gaussian		Two-moon	
	$\sigma = 0.10$	$\sigma = 0.20$	$\sigma = 0.08$	$\sigma = 0.18$
p_{00}	98.58	85.38	95.12	83.54
p_{10}	0.43	-1.19	0.89	5.27
p_{01}	0.03	0.28	-0.76	0.01
p_{20}	-0.25	-0.07	-0.65	-4.38
p_{11}	0.77×10^{-2}	0.36	0.33	-0.29
p_{02}	-0.03	-0.05	0.04	-0.06
p_{30}	0.05	0.27	0.32	1.55
p_{21}	0.03	-0.15	-0.03	0.2.6
p_{12}	-0.85×10^{-2}	-0.99×10^{-2}	-0.07	0.76×10^{-2}
p_{03}	0.48×10^{-2}	0.29×10^{-2}	0.48×10^{-2}	0.42×10^{-2}
p_{40}	-0.58×10^{-2}	-0.07	-0.07	-0.25
p_{31}	-0.55×10^{-2}	0.01	0.01	-6.97×10^{-2}
p_{22}	0.00	0.79×10^{-2}	0.15×10^{-2}	0.39×10^{-2}
p_{13}	0.36×10^{-3}	-0.59×10^{-3}	0.43×10^{-2}	-0.21×10^{-2}
p_{04}	-0.25×10^{-3}	0.00	-0.05×10^{-2}	0.00
p_{50}	0.31×10^{-3}	0.47×10^{-2}	0.62×10^{-2}	1.54×10^{-2}
p_{41}	0.00	0.35×10^{-3}	0.17×10^{-2}	0.49×10^{-2}
p_{32}	0.22×10^{-3}	-0.73×10^{-3}	0.04×10^{-2}	0.05×10^{-2}
p_{23}	0.00	0.00	0.02×10^{-2}	-0.03×10^{-2}
p_{14}	0.00	0.00	0.00	0.00
p_{05}	0.00	0.00	0.00	0.00

Table 5.3: CNN architecture for the Caltech101 (30 classes subset) database.

Layers	Filter Size	Stride	Padding	Output $W \times H \times L$
Input	-	-	-	$32 \times 32 \times 1$
Conv-ReLU	5×5	1	0	$14 \times 14 \times 30$
Avg_pool	2×2	2		
Conv-ReLU	5×5	1	0	$10 \times 10 \times 60$
Avg_pool	2×2	2		
FC-ReLU	-	-	-	200
FC-Softmax	-	-	-	30

size is 190 (5 atoms per class) for LC-RLSDLA. The CNN architecture used for this database is described in Table 4.1 (original images are resized to 32×32 pixels). For the AR database, T_{prop} and U consist of five and 12 images per class, respectively. For SVMs and LC-RLSDLA, the feature descriptors are randomfaces of size $N = 540$.

The dictionary size is 400 (4 atoms per class) for LC-RLSDLA. The CNN architecture used is described in Table 4.2 (original images are converted to greyscale and resized to 50×50 pixels). For Caltech101, T_{prop} and U consist of ten and 30 images per class, respectively. For SVMs and LC-RLSDLA, SIFT descriptors are first extracted. Next, spatial pyramid features, based on the SIFT descriptors, are obtained. The dimension of the spatial pyramid features is then reduced to 3000 through PCA. For LC-RLSDLA, the dictionary size is 300 (10 atoms per class). The CNN architecture used for this database is described in Table 5.3 (original images are converted to greyscale and resized to 32×32 pixels). For SVMs, the one-vs-all approach is used for all the databases.

5.6.3 Results: CPAL-LR for Face and Object Recognition

The performance of CPAL-LR, along with that of the considered active learning approaches, as a function of the number of selected instances N_{AL}, for the different algorithms and databases, is shown in Fig. 5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, and 5.14. It is observed that the performance of the different pattern classification algorithms is significantly improved through active learning, for all the considered databases. Notice that the performance of CPAL-LR compares favorably with that of the other active learning techniques. This demonstrates the effectiveness of the proposed approach.

The results for the Extended YaleB database in Fig. 5.6 (LC-RLSDLA) show that the biggest performance gain is obtained by CPAL-LR for $N_{AL} = 300$. Table 5.4 shows that for $N_{AL} = 500$ the classification accuracy of (rnd), AL(MCLU), AL(CBD), AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), AL(Sparse) is 83.7%, 86.1%, 86.3%, 86.7%, 87.0%, 86.8% and 86.7%, respectively, whereas that of CPAL-LR is 87.3

The results for the AR database in Fig. 5.13 (CNNs) show that the largest performance gain is obtained when CPAL-LR is applied for $N_{AL} = 500$, which is about 9.0%, with respect to random sampling. It can also be seen that the performance of

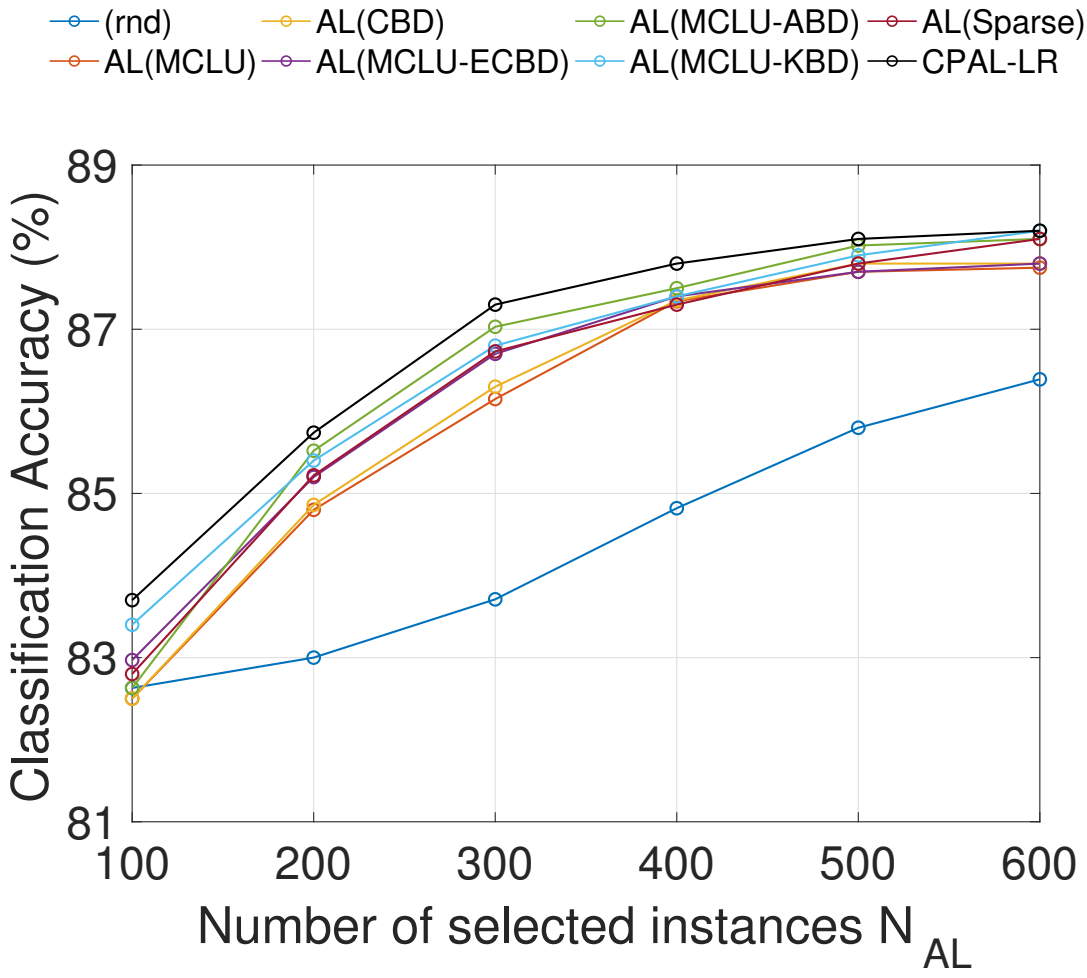


Figure 5.6: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}, YaleB (LC-RLSDLA).

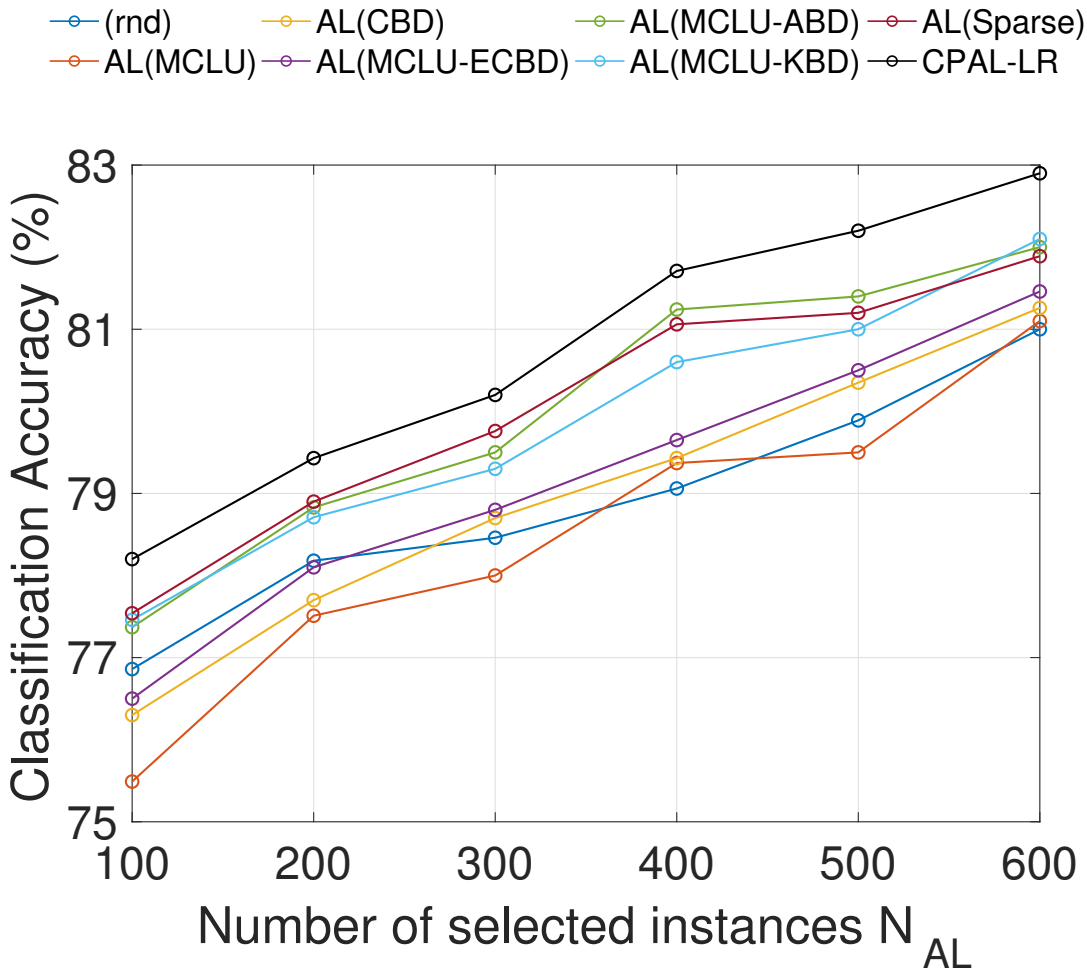


Figure 5.7: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}, AR (LC-RLSDLA).

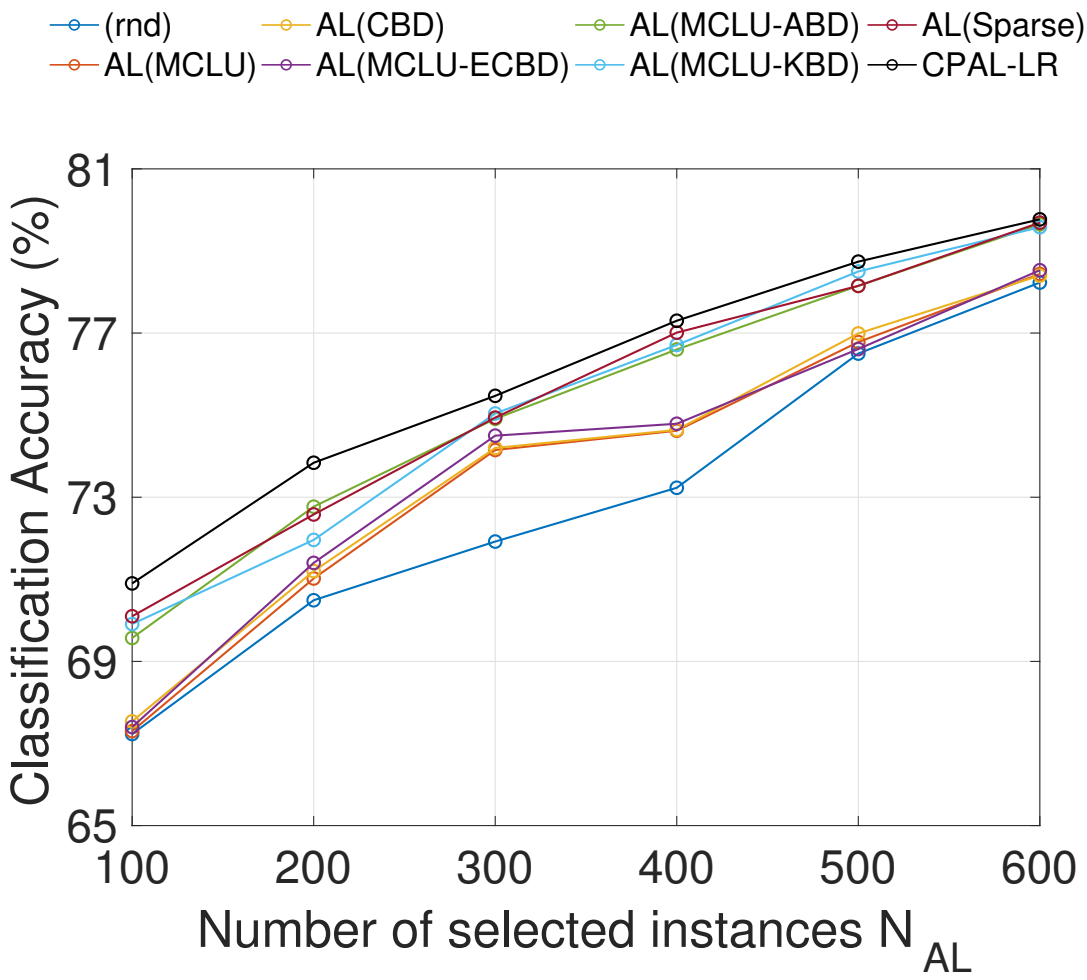


Figure 5.8: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}, Caltech101 (LC-RLSDLA).

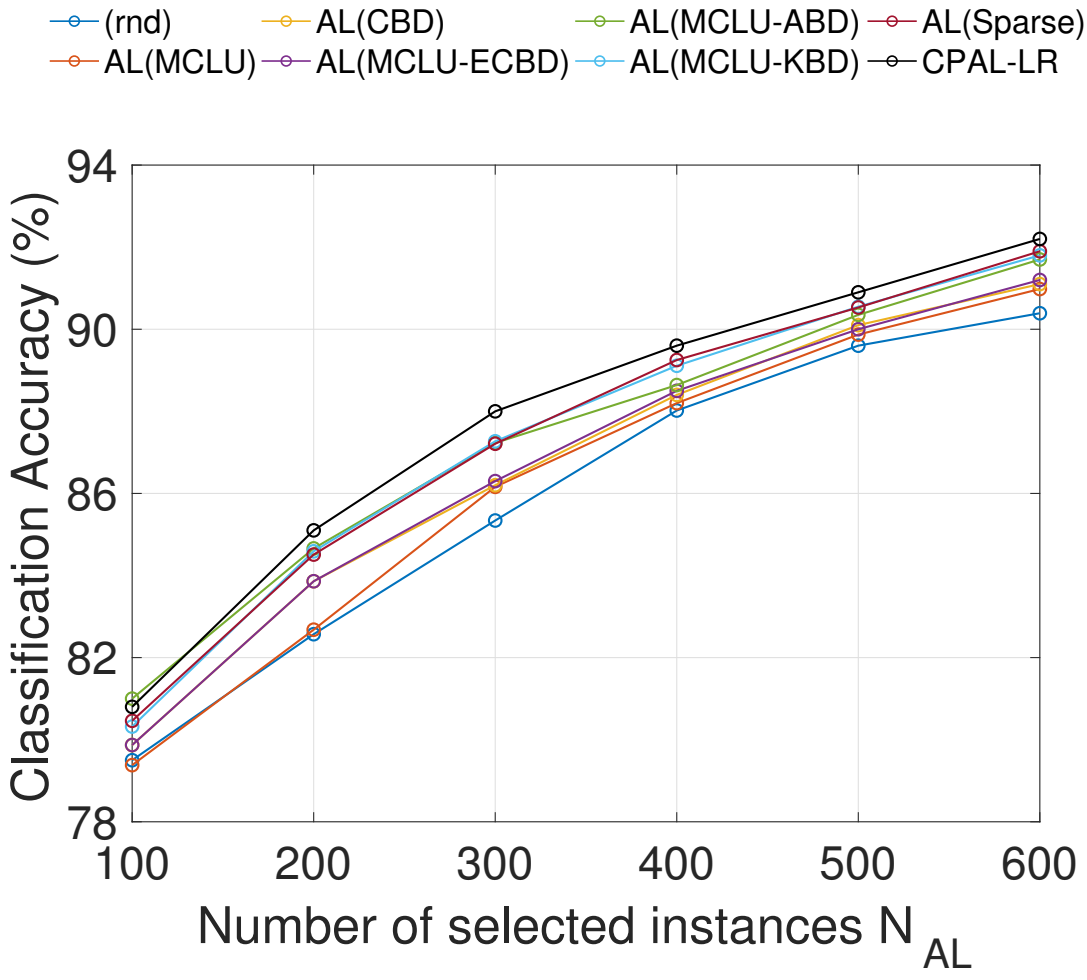


Figure 5.9: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}, YaleB (SVM).

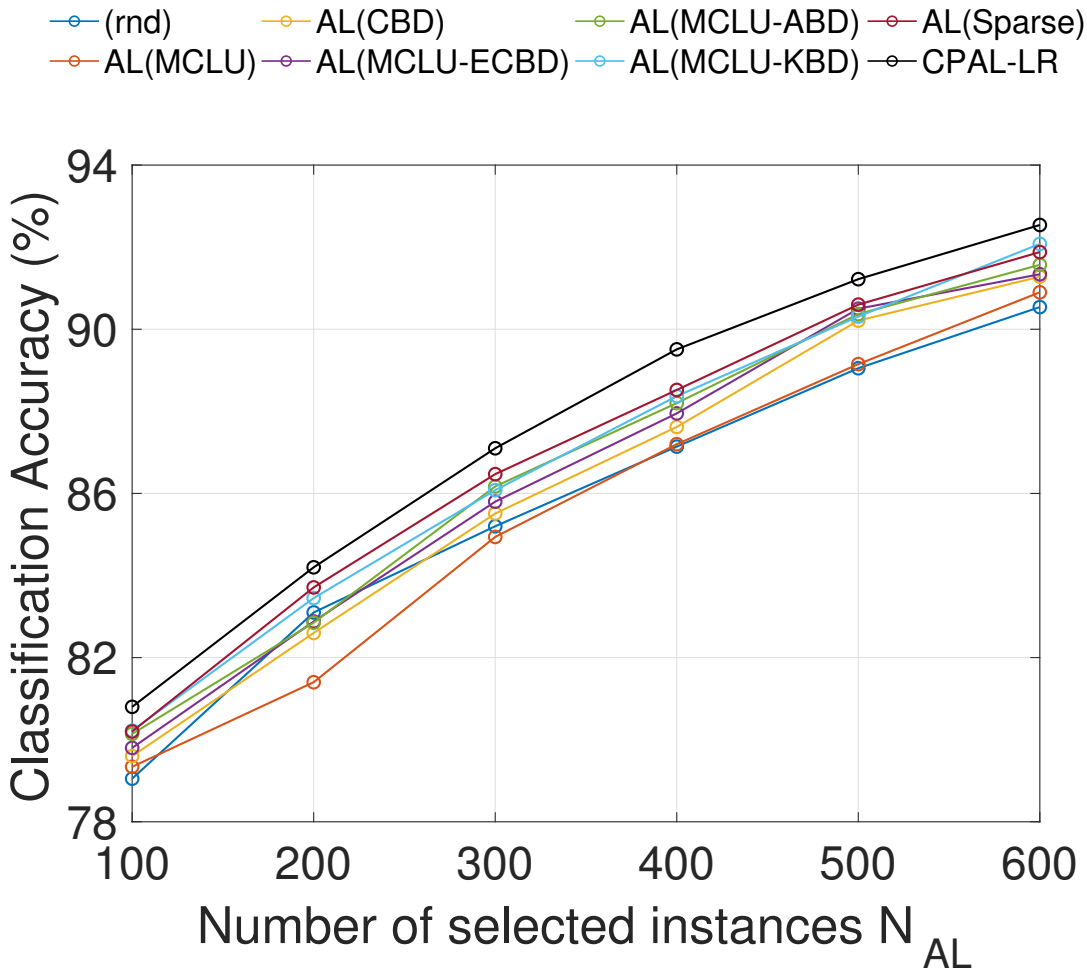


Figure 5.10: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}, AR (SVM).

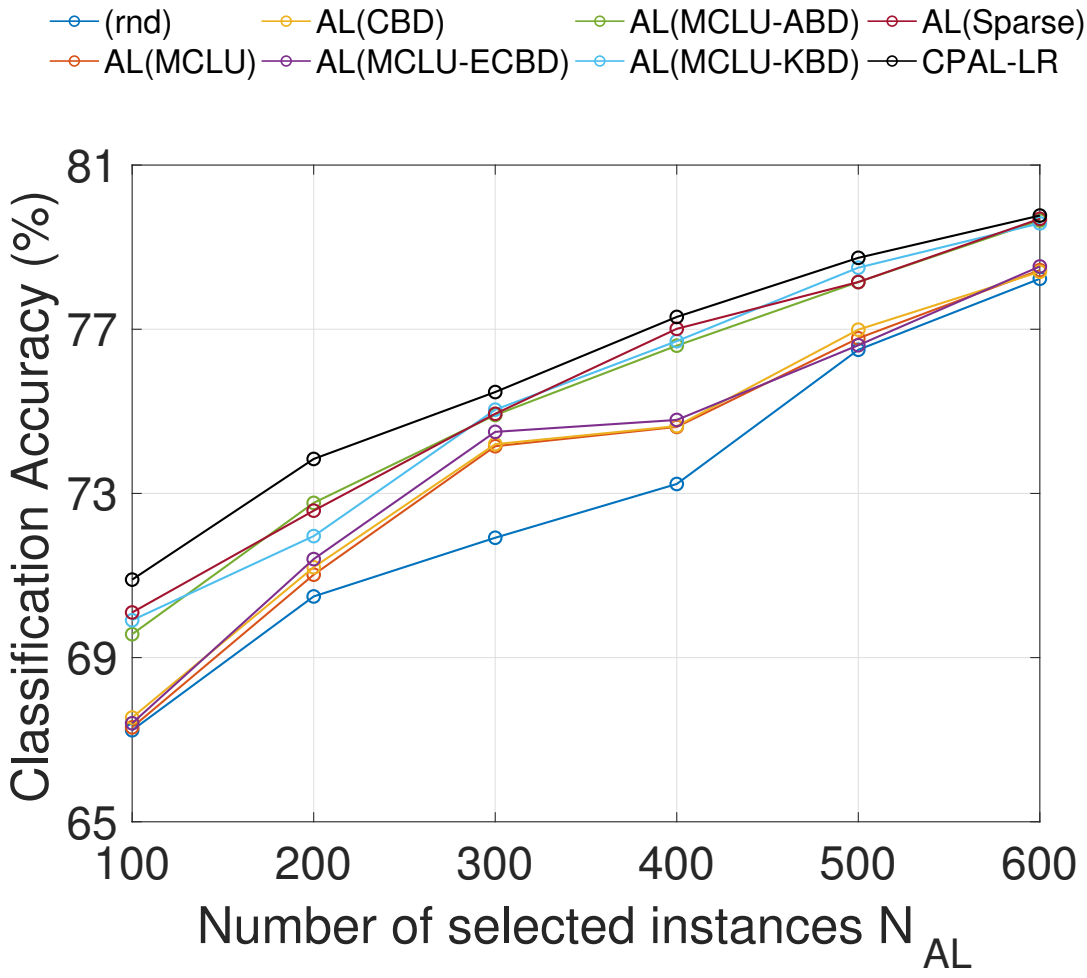


Figure 5.11: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}, Caltech101 (SVM).

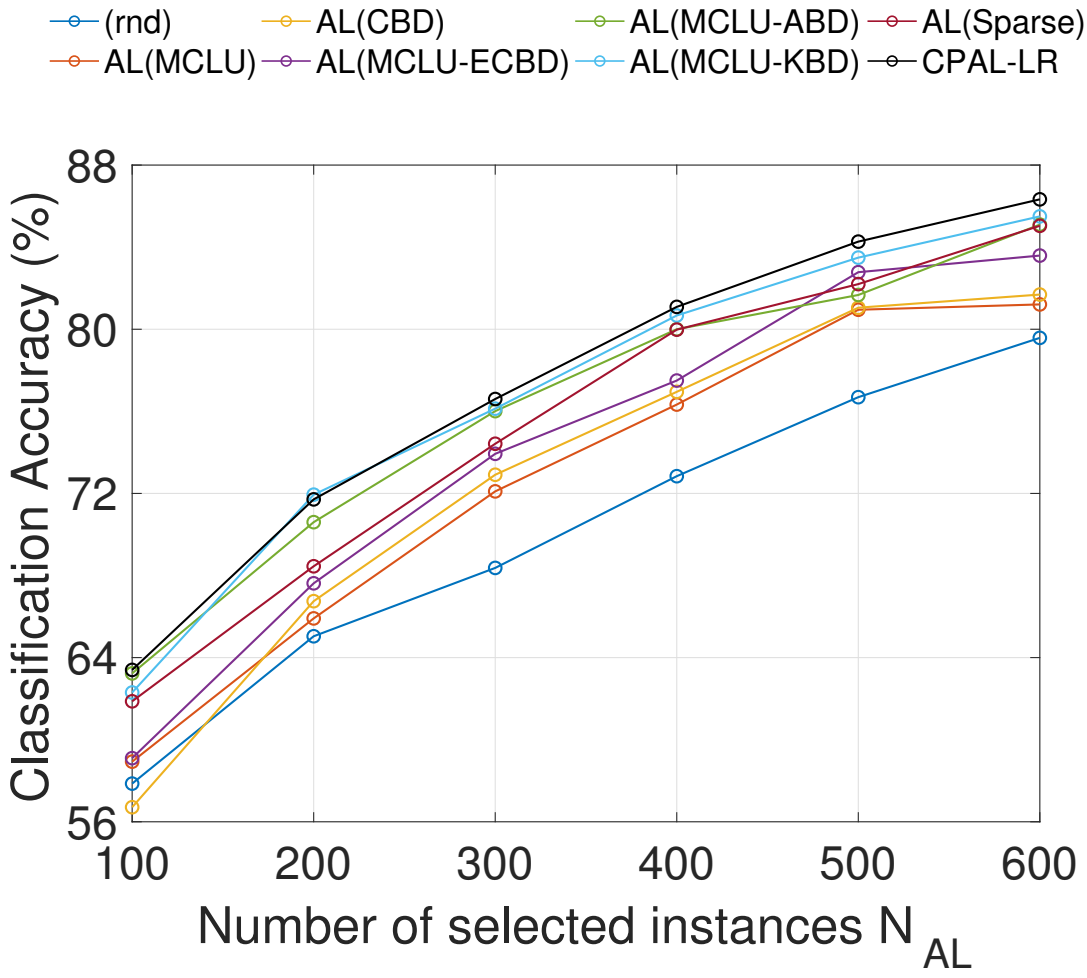


Figure 5.12: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}, YaleB (CNN).

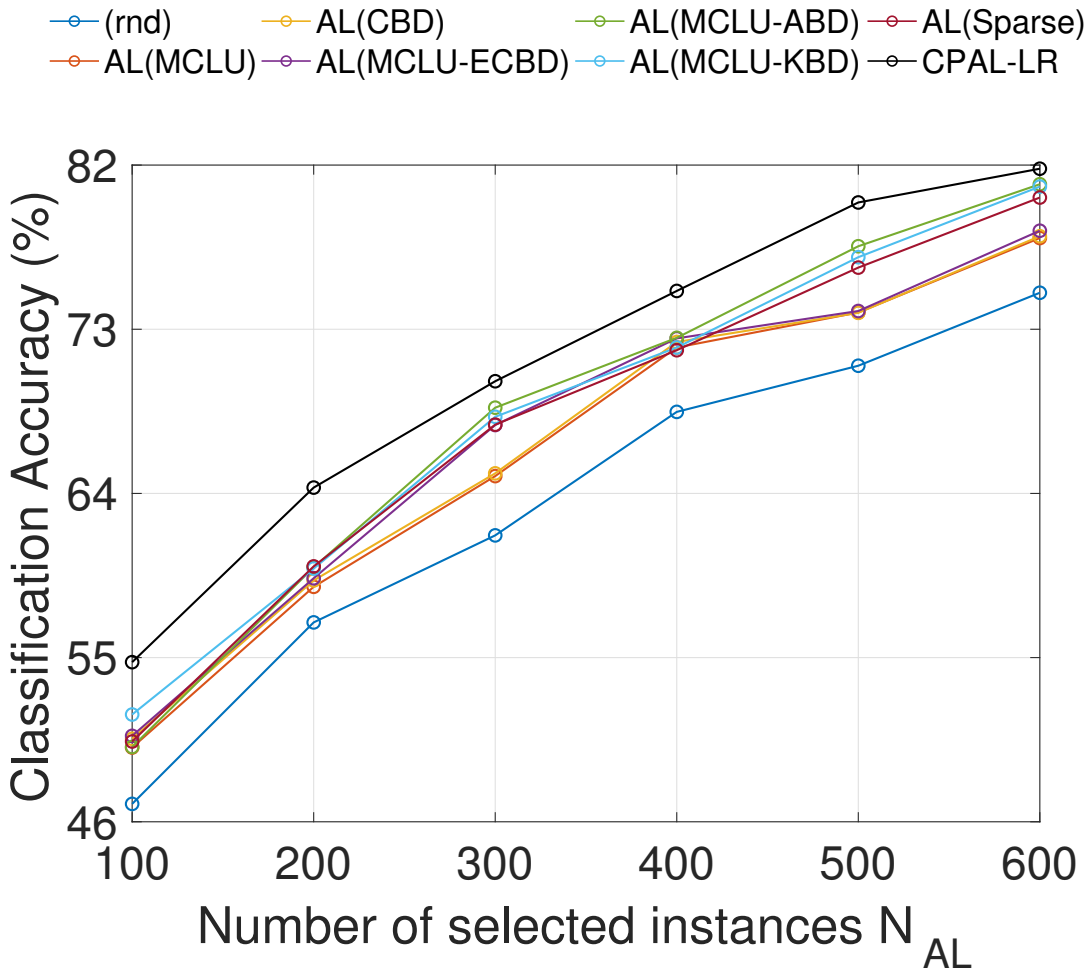


Figure 5.13: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}, AR (CNN).

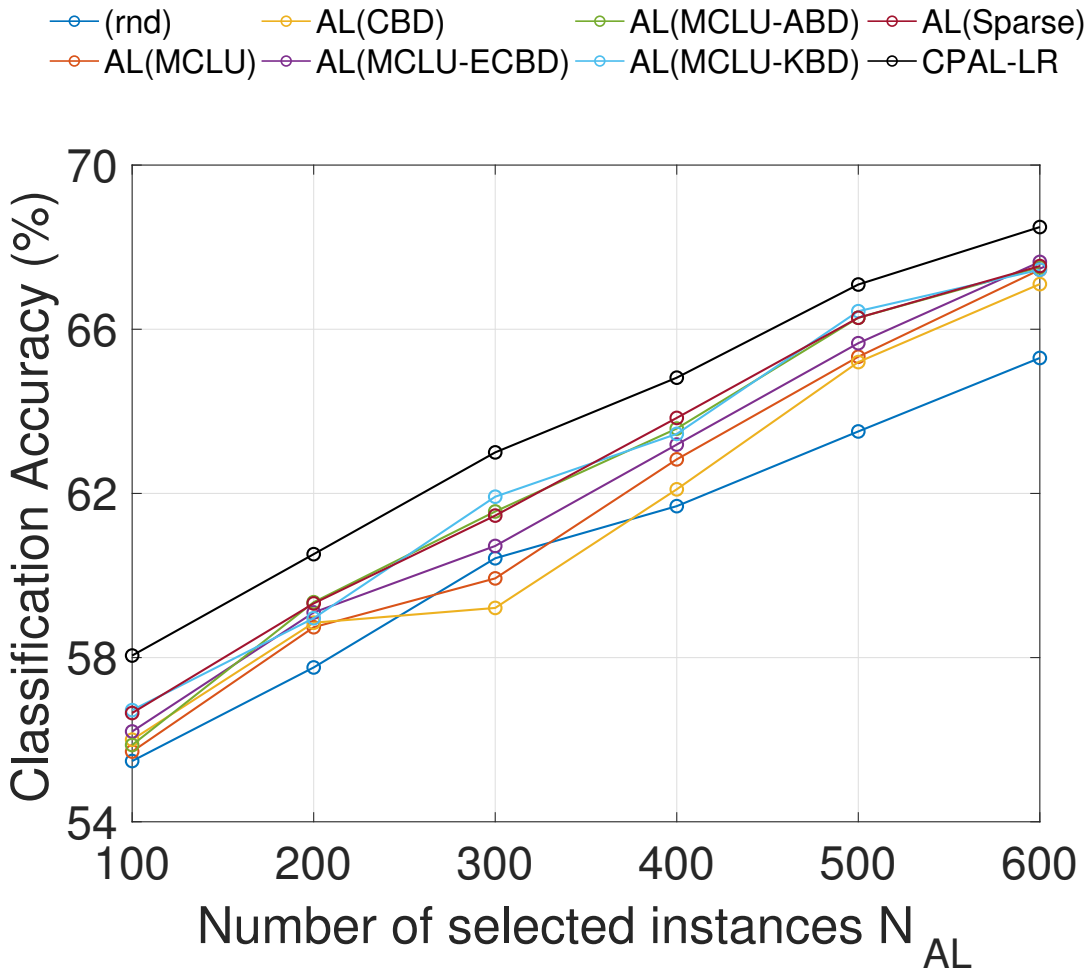


Figure 5.14: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}, Caltech101 (CNN).

Table 5.4: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}.

Algorithm	Query function	YaleB			AR			Caltech101		
		300	400	500	300	400	500	200	300	400
LC-RLSDLA	(rnd)	83.7	84.8	85.8	78.5	79.0	79.9	68.3	69.1	69.9
	AL(MCLU)	86.1	87.3	87.7	78.0	79.4	79.5	69.3	69.2	71.0
	AL(CBD)	86.3	87.3	87.8	78.7	79.4	80.3	69.4	69.5	71.1
	AL(MCLU-ECBD)	86.7	87.4	87.7	78.8	79.6	80.5	69.6	69.9	71.5
	AL(MCLU-ABD)	87.0	87.5	88.0	79.5	81.2	81.4	69.7	70.8	72.0
	AL(MCLU-KBD)	86.8	87.4	87.9	79.3	80.6	81.0	69.5	70.4	71.8
	AL(Sparse)	86.7	87.3	87.8	79.7	81.1	81.2	69.9	70.5	71.7
	CPAL-LR	**87.3**	**87.8**	**88.1**	**80.2**	**81.7**	**82.2**	**70.6**	**71.3**	**72.6**
SVM	(rnd)	85.3	88.0	89.6	85.2	87.1	89.0	70.5	71.9	73.2
	AL(MCLU)	86.2	88.2	89.9	84.9	87.2	89.1	71.0	74.2	74.6
	AL(CBD)	86.2	88.4	90.1	85.5	87.6	90.2	71.2	74.2	74.6
	AL(MCLU-ECBD)	86.3	88.5	90.0	85.8	87.9	90.5	71.4	74.5	74.8
	AL(MCLU-ABD)	87.2	88.6	90.4	86.2	88.2	90.4	72.8	74.9	76.6
	AL(MCLU-KBD)	87.3	89.1	90.5	86.1	88.4	90.3	71.9	75.0	76.7
	AL(Sparse)	87.2	89.2	90.5	86.4	88.5	90.6	72.6	74.9	77.0
	CPAL-LR	**88.0**	**89.6**	**90.9**	**87.1**	**89.5**	**91.2**	**73.8**	**75.5**	**77.3**
CNN	(rnd)	68.4	72.8	76.7	61.7	68.5	71.0	57.8	60.4	61.7
	AL(MCLU)	72.1	76.3	80.9	64.9	72.0	73.9	58.7	59.9	62.8
	AL(CBD)	72.9	76.9	81.0	65.1	72.3	73.9	58.9	59.2	62.1
	AL(MCLU-ECBD)	73.9	77.5	82.8	67.7	72.5	74.0	59.1	60.7	63.2
	AL(MCLU-ABD)	76.0	80.0	81.7	68.7	72.5	77.5	59.4	61.6	63.6
	AL(MCLU-KBD)	76.1	80.7	83.5	68.2	72.0	76.9	59.0	61.9	63.5
	AL(Sparse)	74.4	80.0	82.2	67.7	71.8	76.4	59.3	61.5	63.9
	CPAL-LR	**76.6**	**81.1**	**84.3**	**70.1**	**75.1**	**79.9**	**60.5**	**63.0**	**64.8**

CPAL-LR is highest among all the considered approaches for the different values of N_{AL}. For instance, for $N_{AL} = 500$, the classification accuracy of (rnd), AL(MCLU), AL(CBD), AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), AL(Sparse) is 71.0%, 73.9%, 73.9%, 74.0%, 77.5%, 76.9% and 76.4%, respectively, whereas that of CPAL-LR is 79.9%.

Similar results are obtained for Caltech101, as shown in Fig. 5.11 (SVMs). For instance, for $N_{AL} = 200$, the largest performance improvement is obtained when CPAL-LR is applied, which is about 3.3%, with respect to random sampling. The classification accuracy of (rnd), AL(MCLU), AL(CBD), AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), AL(Sparse) is 70.5%, 71.0%, 71.2%, 71.4%, 72.8%, 71.9%

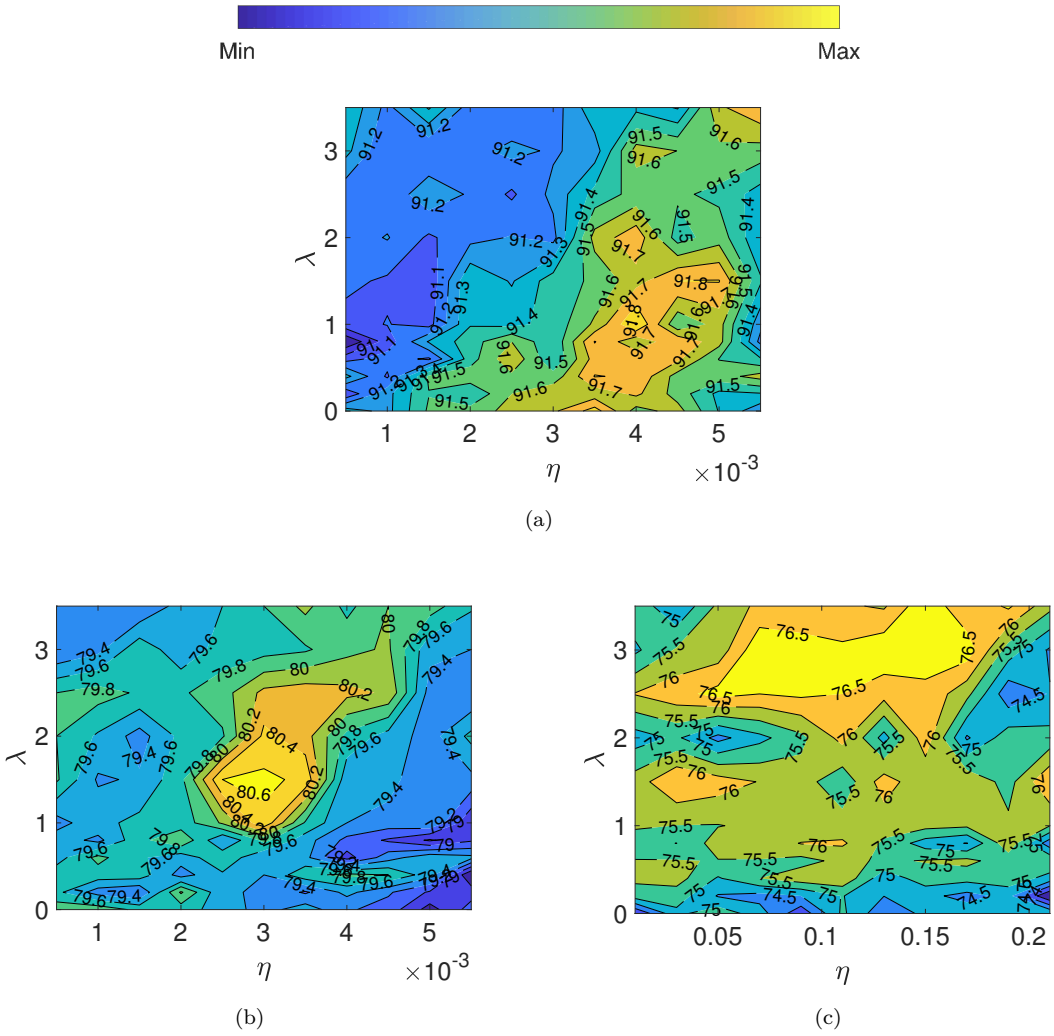


Figure 5.15: Classification accuracy (%) obtained through CPAL-LR (SVMs) as a function of η and λ, (a) YaleB ($N_{AL} = 600$), (b) AR ($N_{AL} = 100$), (c) Caltech101 ($N_{AL} = 400$).

and 72.6%, respectively, whereas that of CPAL-LR is 73.8%.

Figure 5.15 shows the classification accuracy of CPAL-LR (SVMs) as a function of the parameters η and λ for the Extended YaleB, AR, and Caltech101 databases (average over the five trials for the different values of η and λ). It is observed that the best performance is obtained for a combination of uncertainty, diversity, and representativeness, *i.e.*, $\eta, \lambda \geq 0$, for all the considered databases. For the Extended YaleB

database ($N_{AL} = 600$), the best performance is obtained in the region $\lambda \in [0, 2]$ and $\eta \in [3.0 \times 10^{-3}, 5.0 \times 10^{-3}]$, and the classification accuracy peaks when $\eta = 4.0 \times 10^{-3}$ and $\lambda = 1.0$ (91.9%). For the AR database ($N_{AL} = 100$), the best performance is obtained in the region $\lambda \in [1, 2]$ and $\eta \in [2.0 \times 10^{-3}, 4.0 \times 10^{-3}]$, and the classification accuracy peaks when $\eta = 3.0 \times 10^{-3}$ and $\lambda = 1.5$ (80.7%). For Caltech101 ($N_{AL} = 200$), the best performance is obtained in the region $\lambda \in [2.5, 3.5]$ and $\eta \in [5.0 \times 10^{-2}, 1.0 \times 10^{-1}]$, and the classification accuracy peaks when $\eta = 7.0 \times 10^{-2}$ and $\lambda = 3.0$ (76.9%). Similar results are obtained for LC-RLSDLA and CNNs.

The results on face and object recognition show that CPAL-LR improves the performance of several pattern classification algorithms across different databases, outperforming other state-of-the-art active learning techniques.

5.6.4 Results: Quality of CPAL-LR Confidence Values

In this section, the quality of the confidence values produced by CPAL-LR (through Algorithm 4) is compared with that of the confidence values obtained through the hinge and margin nonconformity measures, which are given by $A_{hinge}^{(\mathcal{H}_q)} := 1 - \max_{i=1,\ldots,M,} o_j^{(i)}$, and $A_{margin}^{(\mathcal{H}_q)} := -o_j^{(q)} + \max_{i=1,\ldots,M, i \neq q} o_j^{(i)}$, respectively. Notice that the hinge and margin nonconformity measures are particular cases of the proposed nonconformity measure described by equation (4.1), when $\gamma = 1.0$ and $\gamma = 0.5$, respectively. Experiments are performed for SVMs, LC-RLSDLA, and CNNs on the Extended YaleB, AR, and Caltech101 databases. Different significance levels, $\epsilon \in [0, 1]$, are used yielding different prediction sets Ψ_{n+j}^{ϵ}, for test instances $\mathbf{x}_{n+j}$. The quality of the CPAL-LR confidence values is demonstrated using the three metrics described in Section 4: ValeE, SinP, and AvgC.

Figure 5.16 shows the performance of the proposed nonconformity measure, as a function of the parameter $\gamma \in [0, 1]$, using the three aforementioned metrics, for $\epsilon = 0.1$. It is observed in Fig. 5.16(a) that ValE fluctuates around 10%, for the different values of λ, across all the considered pattern classification algorithms and databases, which agrees with the validity property (ValE $\approx$ 10%). The parameter γ can be adjusted to

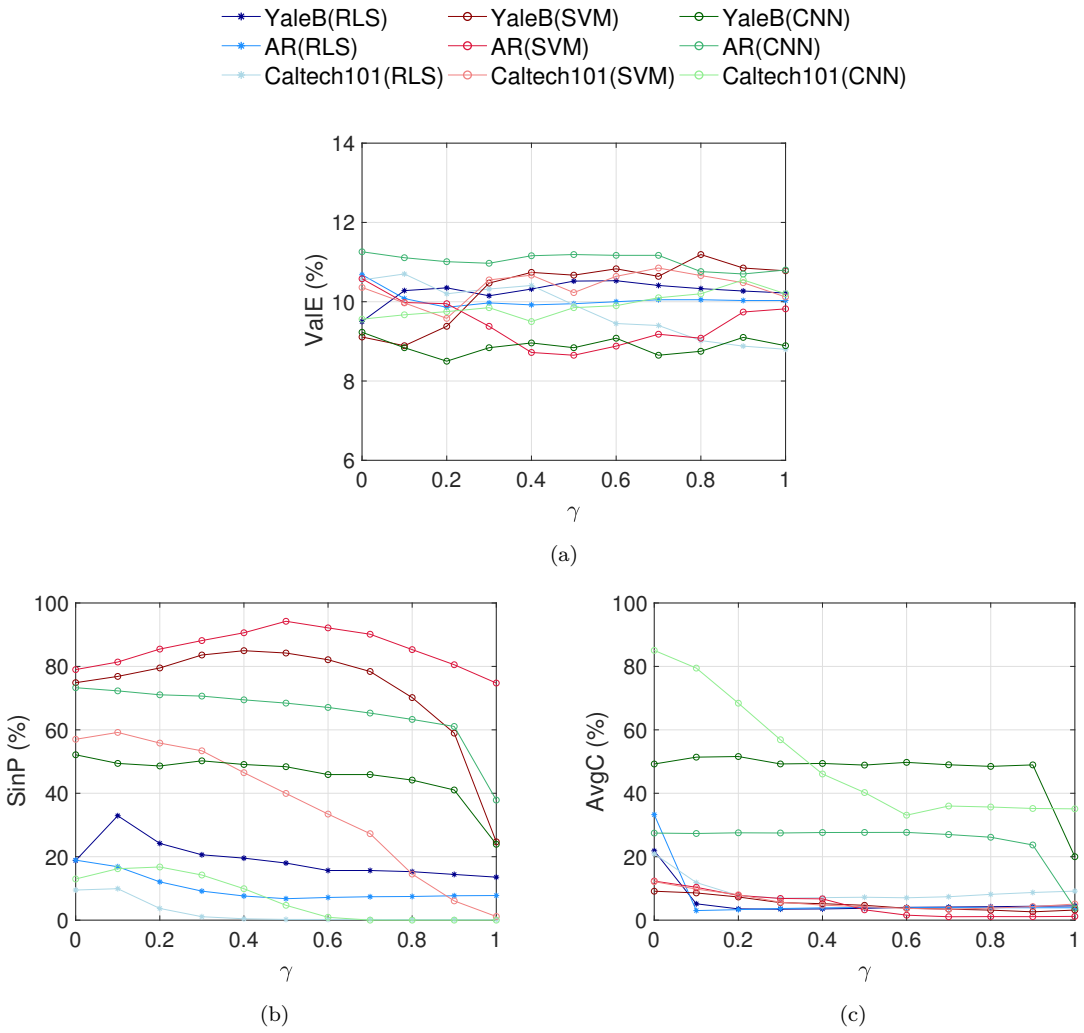


Figure 5.16: Performance of the proposed nonconformity measure for $\epsilon = 0.1$ as a function of the parameter $\gamma \in [0, 1]$ using different metrics: (a) ValE, (b) SinP, (c) AvgC.

obtain the desired performance . For instance, when $\gamma = 0.3$ (LC-RLSDLA, YaleB), ValeE=10.1%. For SVMs (Caltech101, $\gamma = 0.1$), ValeE=10.0%. For CNNs (AR, $\gamma = 0.9$), ValeE=10.7%. This demonstrates the usefulness of the CPAL-LR confidence values.

Fig. 5.16(b) shows the behavior of singleton predictions as a function of γ (SinP). It is observed that SinP behaves differently across the considered pattern classification

Table 5.5: Performance of hinge, margin, and CPAL-LR nonconformity measures.

Algorithm	Database	Confidence (%)	Performance (%)								
			Hinge			Margin			CPAL-LR		
			ValE	SinP	AvgC	ValE	SinP	AvgC	ValE	SinP	AvgC
LC-RLSDLA	YaleB	98	2.9	0.0	28.5	2.5	0.0	27.5	**2.0**	**3.8**	**24.2**
		95	5.8	0.9	10.8	5.7	2.4	8.6	**5.0**	**11.6**	**7.6**
		90	10.2	13.5	4.5	10.5	18.0	3.7	**10.1**	**32.9**	**3.4**
	AR	98	**2.7**	0.0	**26.0**	2.8	0.0	27.3	**2.7**	**2.0**	**26.0**
		95	5.4	0.0	15.8	5.4	0.0	15.8	**5.3**	**8.7**	**15.3**
		90	**10.0**	7.7	3.9	9.9	6.8	4.0	**10.0**	**18.9**	**3.0**
	Cal101	98	2.4	0.0	21.1	2.7	0.0	20.5	**2.2**	**3.7**	**19.9**
		95	5.3	0.0	15.2	4.9	0.2	**11.9**	**5.0**	**6.4**	**11.9**
		90	8.8	0.2	9.2	**9.9**	0.2	7.2	**9.9**	**9.9**	**6.9**
SVM	YaleB	98	1.8	0.4	22.2	2.5	56.2	13.9	**2.2**	**63.4**	**12.4**
		95	4.9	3.8	**7.7**	5.4	69.4	9.5	**5.0**	**72.5**	**7.7**
		90	10.8	24.7	3.2	10.7	84.2	4.7	**10.5**	**84.9**	**3.1**
	AR	98	2.4	10.2	6.7	1.8	**85.3**	11.4	**2.0**	**85.3**	**6.1**
		95	4.6	21.1	**2.9**	4.5	**88.9**	7.5	**5.0**	**88.9**	**2.9**
		90	9.8	74.7	1.2	8.6	94.2	3.2	**10.0**	**94.2**	**1.0**
	Cal101	98	2.5	0.0	16.3	2.7	25.5	**8.8**	**2.3**	**46.3**	**8.8**
		95	5.1	0.4	12.3	5.5	32.0	**6.3**	**5.0**	**50.1**	**6.3**
		90	10.1	1.2	5.1	10.2	39.9	4.2	**10.0**	**59.2**	**3.4**
CNN	YaleB	98	2.6	8.5	**44.1**	2.2	25.4	73.1	**2.0**	**28.3**	**44.1**
		95	5.1	9.9	**40.8**	**5.0**	42.4	55.8	**5.0**	**47.1**	**40.8**
		90	8.9	23.9	**20.0**	8.8	48.4	48.9	**9.2**	**52.1**	**20.0**
	AR	98	**2.1**	9.5	**16.1**	1.7	36.8	59.5	**2.1**	**39.9**	**16.1**
		95	**4.3**	18.9	**8.8**	6.2	58.5	38.0	**4.3**	**62.1**	**8.8**
		90	10.8	37.9	**3.8**	11.2	68.4	27.6	**10.7**	**73.3**	**3.8**
	Cal101	98	2.8	0.0	71.1	2.5	0.4	**63.0**	**1.7**	**2.9**	**63.0**
		95	6.5	0.0	56.8	5.9	0.8	54.1	**5.2**	**6.3**	**46.8**
		90	10.2	0.0	35.1	9.8	2.0	40.2	**10.1**	**13.3**	**33.1**

algorithms. The results in Fig. 5.16(b) show that SVMs obtain the highest number of singleton predictions, followed by CNNs and LC-RLSDLA, respectively. For the Extended YaleB database, the production of singleton predictions peaks when $\gamma = 0.1$, $\gamma = 0.4$, and $\gamma = 0.2$ for LC-RLSDLA (32.9%), SVMs (84.9%), and CNNs (52.1%), respectively.

The average number of class labels in the prediction sets, as a percentage of the total number of classes (AvgC), is shown in Fig. 5.16(c), for different values of γ. The results show that LC-RLSDLA and SVMs produce more discriminative sets Ψ^{ϵ} than CNNs (low AvgC). For the AR database, AvgC reaches its minimum when $\gamma = 0.1$, $\gamma = 0.7$, and $\gamma = 1.0$ for LC-RLSDLA (3.0%), SVMs (1.0%), and CNNs (3.8%), respectively.

The performance results of the hinge, margin, and CPAL-LR nonconformity measures are summarized in Table 5.5. For CPAL-LR, the best results are shown (from those obtained using different values of γ). Table 5.5 shows that CPAL-LR achieves similar or better performance than that obtained through the hinge and margin nonconformity measures, for the considered performance metrics.

5.7 Chapter Conclusion

A conformal prediction based active learning algorithm is presented in this chapter. The proposed approach uses a novel query function that determines the relevance of unlabeled instances through the solution of a constrained linear regression model, incorporating uncertainty, diversity, and representativeness in the optimization problem.

CPAL-LR is implemented in conjunction with three different pattern classification algorithms: SVMs, sparse coding (LC-RLSDLA), and CNNs. Experiments conducted on face and object recognition databases show that CPAL-LR outperforms previous work on active learning, improving performance across different pattern classification techniques and databases. Moreover, experiments performed on synthetic data provide a greater insight into the working mechanism of the proposed approach and also show the behavior of the parameters η and λ for two different synthetic databases.

In addition to performance enhancement, CPAL-LR produces reliable confidence values that are used to predict class labels with guaranteed error rate. Experimental results show that the proposed nonconformity measure achieves similar or better performance (quality of the confidence values) than previously proposed techniques.

In the following chapter, a nonlinear constrained optimization problem is considered to determine the relevance of instances in the unlabeled pool. Experiments conducted on synthetic databases are performed to give a greater insight into the process of instance selection. Moreover, active learning is performed on a video database for emotion recognition.

Chapter 6

CONFORMAL PREDICTION BASED ACTIVE LEARNING BY NONLINEAR CONSTRAINED OPTIMIZATION

We propose a conformal prediction based active learning algorithm, referred to as CPAL-NCO. The proposed approach uses a novel query function that considers uncertainty, diversity, and representativeness as the selection criteria. In the remainder of this section, the proposed query function is introduced, and the CPAL-NCO algorithm is described.

6.1 CPAL-NCO Query Function

The proposed query function determines the relevance of unlabeled instances through the solution of quadratic programming optimization problem, incorporating uncertainty, diversity, and representativeness. Define $U = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L\}$ as the unlabeled pool. Let $T_{prop} = \{z_1, ..., z_\ell\}$ be the proper training set, containing ℓ instances with their corresponding class labels. Let T_d be a set containing the instances that have been selected from the unlabeled pool. The relevance of an instance is given by the following expression:

$$r(T_d) = (1 - \alpha - \beta) \sum_{i \in T_d} u_i + \alpha \sum_{i \in T_d} \rho_i^{(k_1)} - \beta \sum_{i \in T_d} \rho_i^{(k_2)}, \tag{6.1}$$

where the first and second terms in (6.1), with their corresponding weights $\{\alpha, \beta \in [0, 1] \mid \alpha + \beta <= 1\}$, promote diversity and representativeness, respectively. The term u_i is the uncertainty of instance $\mathbf{x}_i$, calculated according to equation (2.3), and $k_1 \ll k_2$.

The term $\rho_i^{(k)}$ is computed using the distance of instance $\mathbf{x}_i$ from its k-nearest neighbors in the unlabeled pool, denoted as $\mathbf{z}_i^{(n)}$ $(n = 1, \ldots, k)$, as:

$$\hat{\rho}_i^{(k)} = \sum_{n=1}^{k} \left\| \mathbf{x}_i - \mathbf{z}_i^{(n)} \right\|_2^2. \tag{6.2}$$

Notice that the value $\hat{\rho}_i^{(k)}$ will be low if instance $\mathbf{x}_i$ is close to its k-nearest neighbors (densely populated region). Conversely, the value $\hat{d}_i$ will be high if instance $\mathbf{x}_i$ is far from its k-nearest neighbors (sparsely populated region). Define $\rho_{max} = \max\{\rho_i^{(k)}\}$. CPAL-NCO computes the values $\rho_i^{(k)}$ for all instances $\mathbf{x}_i$ in the unlabeled pool $(i = 1, \ldots, |U|)$ as:

$$\rho_i^{(k)} = \hat{\rho}_i^{(k)} / \rho_{max}. \tag{6.3}$$

The term $\rho_i^{(k_1)}$ $(k_1 \ll k_2,)$ indicates how close instance $\mathbf{x}_i$ is to other instances in a small neighborhood. Therefore, low values of $\rho_i^{(k_1)}$ are penalized to promote diversity. The term $\rho_i^{(k_1)}$ indicates how representative of the data instance $\mathbf{x}_i$ is by measuring the distance of $\mathbf{x}_i$ to other instances in a large neighborhood. High values of $\rho_i^{(k_1)}$ are penalized to promote representativeness.

CPAL-NCO selects a batch T_d of unlabeled instances so as to maximize (6.1). Since brute force search methods are prohibitive, numerical optimization techniques are employed to obtain a solution. Following the scheme utilized in [24, 26], we define a binary vector $\mathbf{v} = [v_1, ..., v_{|U|}]$, where each entry denotes whether the corresponding point is to be queried for its class label. The objective function given by (6.1) can be rewritten in terms of $\mathbf{v}$ as:

$$\max_{\mathbf{v}} \ (1 - \alpha - \beta) \sum_{i \in U} u_i v_i + \alpha \sum_{i \in U} \rho_i^{(k_1)} v_i - \beta \sum_{i \in U} \rho_i^{(k_2)} v_i \tag{6.4}$$
$$\boldsymbol{s.t.}\ v_i \in \{0, 1\},$$

The above optimization is an integer programming problem and is NP hard. We

therefore relax the constraints to make it a continuous optimization problem:

$$\max_{\mathbf{v}} \ (1-\alpha-\beta)\sum_{i\in U} u_i v_i + \alpha \sum_{i\in U} \rho_i^{(k_1)} v_i - \beta \sum_{i\in U} \rho_i^{(k_2)} v_i \tag{6.5}$$
$$\boldsymbol{s.t} \ \ 0 \leq v_i \leq 1$$
$$\mathbf{v}^T\mathbf{1} = N_{AL}$$

The optimization problem in (6.5) is solved as a nonlinear constrained optimization problem using the interior-point barrier method [71, 72]. After solving for vector $\mathbf{v}$, the instances associated with top largest N_{AL} entries in $\mathbf{v}$ are selected from the unlabeled pool.

6.2 CPAL-NCO Nonconformity Measure

Nonconformity measures produce nonconformity scores, which are then used to compute informativeness, as described in Section 2. CPAL-NCO uses the nonconformity measure given by equation (4.1).

6.3 CPAL-NCO Algorithm

We propose an active learning algorithm within the CP framework. First, we split the training set, $T_{train} = \{z_1, ..., z_n\}$, into the proper training set, $T_{prop} = \{z_1, ..., z_\ell\}$, and the calibration set, $T_{cal} = \{z_{\ell+1}, ..., z_{\ell+r}\}$, where $n = \ell + r$, as described in Section 2. Then, the classification rule, C_{prop}, is obtained through the underlying algorithm employing T_{prop}.

The nonconformity scores of the instances in calibration set, T_{cal}, and the unlabeled pool, U, are computed using equation (4.1) and C_{prop}. The nonconformity scores are used to measure the p-values and the uncertainty of instances in the unlabeled pool, according to equation (2.1) and (2.3), respectively.

The terms $\rho_i^{k_1}$ and $\rho_i^{k_2}$ are computed using the k-nearest neighbors approach, according to equations (6.2) and (6.3). Then, the optimization problem described by

Algorithm 5 CPAL-NCO

1: **Input:** Proper training set $T_{prop} = \{z_1, ..., z_\ell\}$, calibration set $T_{cal} = \{z_{\ell+1}, ..., z_{\ell+r}\}$, unlabeled pool $U = \{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+v}\}$, classification rule C_{prop}, number of desired instances N_{AL}, and number of class labels M

2: Compute the weights $p_i^{(k_1)}$ and $p_i^{(k_2)}$, using equations (6.2), and (6.3), for all instances in the unlabeled pool U

3: Use Equation (4.1) and the classification rule C_{prop} to calculate:

- The nonconformity scores $\{\alpha_{\ell+1}, \ldots, \alpha_{\ell+r}\}$ corresponding to the instances in the calibration set.
- The nonconformity scores $\{\alpha_{n+1}^{\mathcal{H}_i}, \ldots, \alpha_{n+v}^{\mathcal{H}_i}\}$ corresponding to the instances in the unlabeled pool, where $i = \{1, \ldots, M\}$

4: Use Equation (2.1) to calculate the p-values associated with the instances in U, and obtain their uncertainty u_{n+j} through equation (2.3), where $j \in \{1, \ldots, v\}$

5: Solve the nonlinear contrained optimization problem in (6.5) and form the set T_d containing the N_{AL} instances from U, with their corresponding class labels, whose associated relevance v is the highest

6: Construct $T_{AL} = T_{prop} \cup T_d$

7: **Output:** T_{AL}

equation (6.5) is solved to obtain the relevance $\mathbf{v}$ of the instances in the unlabeled pool. The N_{AL} instances $\mathbf{x}_i$ whose relevance is highest are selected.

CPAL-NCO returns the training set $T_{AL} = T_{prop} \cup T_d$, where T_d is the set of pairs containing the N_{AL} instances from U, with their corresponding class labels, whose associated relevance $\mathbf{v}$ is the highest after solving the optimization problem in (6.5). The proposed approach is summarized in Algorithm 5.

6.4 CPAL-NCO as a Conformal Predictor

The proposed nonconformity measure, described by equation (4.1), can be used to produce confidence values associated with new predictions, during the testing phase. After training the underlying algorithm and obtaining a classification rule, denoted as C_{train}, the nonconformity scores $\alpha_{n+j}^{(\mathcal{H}_q)}$ and p-values $p(\alpha_{n+j}^{(\mathcal{H}_q)})$, associated with a new instance $\mathbf{x}_{n+j}$, are computed according to equations (4.1) and (2.1), respectively. Then, for a given significance level $\epsilon \in [0, 1]$, we form a set of labels $\Psi_{n+j}^{\epsilon} = \{i : p(\alpha_{n+j}^{(\mathcal{H}_i)}) > \epsilon\}$ containing the correct class label for $\mathbf{x}_{n+j}$ with probability $(1 - \epsilon)$, according to the validity property. CPAL-NCO as a conformal predictor is described in Algorithm 6.

Algorithm 6 CPAL-NCO (conformal predictor)

1: **Input:** Testing instance $\mathbf{x}_{n+j}$, calibration set nonconformity scores $\{\alpha_{\ell+1}, \ldots, \alpha_{\ell+r}\}$, classification rule C_{train}, significance level ϵ, parameter γ, and number of class labels M

2: Use Equations (2.1) and (4.1), along with the classification rule C_{train}, to calculate:

- The nonconformity scores $\alpha_{n+j}^{\mathcal{H}_i}$ corresponding to the new instance $\mathbf{x}_{n+j}$, for the different null hypothesis $\mathcal{H}_i$ $(i = \{1, \ldots, M\})$
- The p-values $p(\alpha_{n+j}^{(\mathcal{H}_i)})$, associated with $\alpha_{n+j}^{\mathcal{H}_i}$

3: Construct the set $\Psi_{n+j}^{\epsilon} = \{i : p(\alpha_{n+j}^{(\mathcal{H}_i)}) > \epsilon\}$

4: **Output:** Ψ_{n+j}^{ϵ}

6.5 Experimental Results

The focus of CPAL-NCO is twofold: 1) to improve the performance of pattern classification algorithms through active learning; and 2) to produce reliable confidence values. Therefore, our goal is to evaluate CPAL-NCO based on the improvement achieved in classification performance and the quality of the produced confidence values. This section is organized as follows. First, we present simulation results obtained on several synthetic databases to provide a greater insight into the proposed query function and show its effectiveness. Then, we evaluate the performance of CPAL-NCO on face and object recognition databases, providing a comparison between the proposed technique and previous work on active learning.

6.5.1 Synthetic Database Experiments

Experiments are conducted on the Gaussian and Two-moon synthetic databases described in Section 5.6.

SVMs are employed for these experiments, using the one-vs-all (OVA) approach. Linear SVMs are used for the Gaussian database, whereas kernel SVMs (polynomial of order 3) are used for the Two-moon database. We compare the performance improvement obtained through CPAL-NCO with that of the following batch active learning approaches: random sampling, *i.e.*, we take instances from the unlabeled pool at random, active learning based on uncertainty [9, 12, 13], clustering [22], clustering with uncertainty [22], uncertainty and ABD [14], uncertainty and KBD [28], and generalized

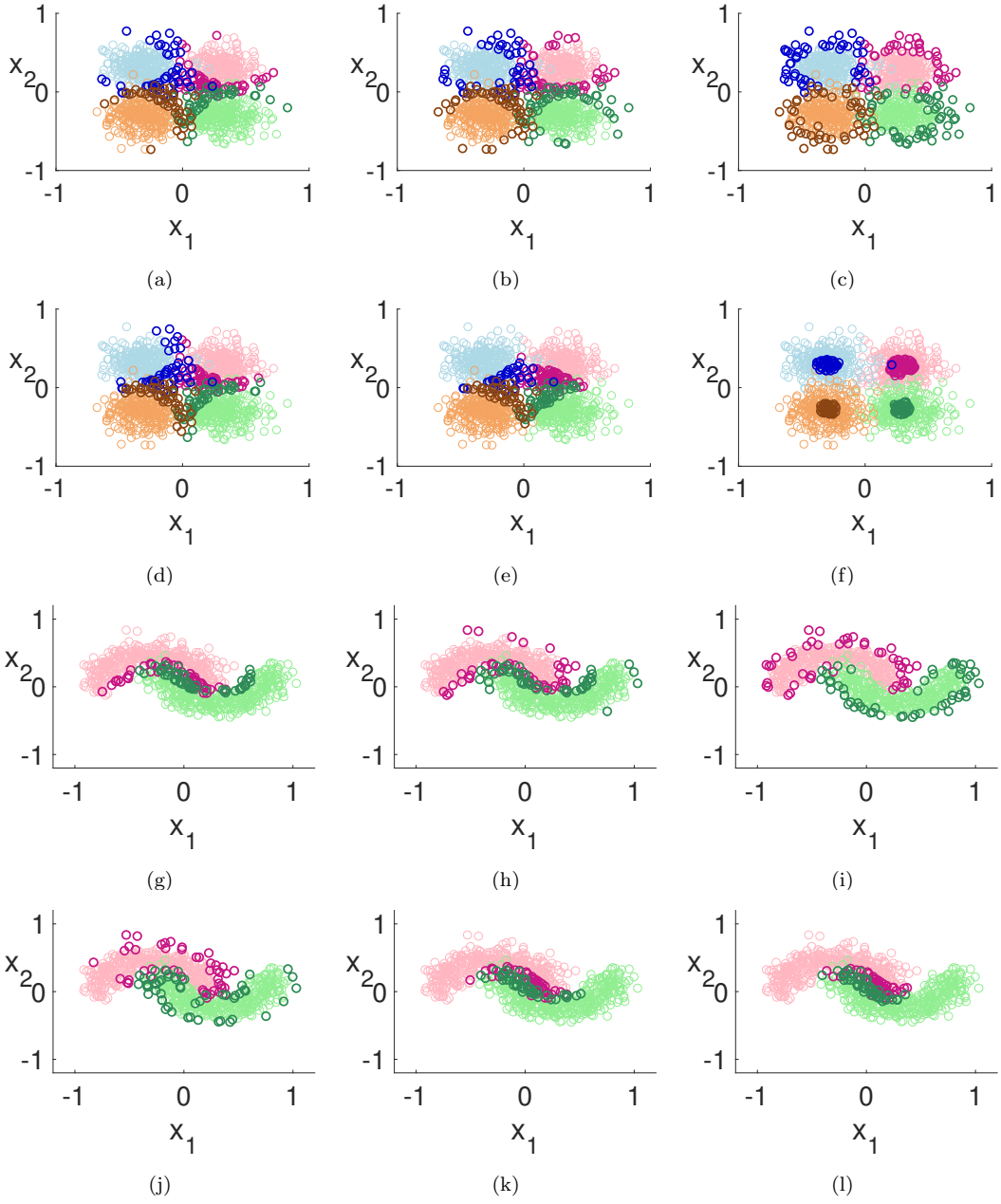


Figure 6.1: Synthetic databases and selected instances (highlighted) (Gaussian $\rightarrow$ $\sigma = 0.14$, and Two-moon $\rightarrow$ $\sigma = 0.10$) using CPAL-NCO (a) ($\alpha = 0$, $\beta = 0$), (b) ($\alpha = 0$, $\beta = 6$), (c) ($\alpha = 0$, $\beta = 1$), (d) ($\alpha = 2$, $\beta = 0$), (e) ($\alpha = 6$, $\beta = 0$), (f) ($\alpha = 1$, $\beta = 0$), (g) ($\alpha = 0$, $\beta = 0$), (h) ($\alpha = 0$, $\beta = 8$), (i) ($\alpha = 0$, $\beta = 1$), (j) ($\alpha = 2$, $\beta = 8$), (k) ($\alpha = 6$, $\beta = 2$), (l) ($\alpha = 1$, $\beta = 0$).

batch mode active learning [26], which are denoted as (rnd), AL(MCLU), AL(CBD),

Table 6.1: Classification accuracy (%) for different query functions and standard deviation σ as a function of the number of selected instances N_{AL}.

Algorithm	Query function	Gaussian				Two-moon			
		$\sigma = 0.13$		$\sigma = 0.17$		$\sigma = 0.08$		$\sigma = 0.14$	
		N_{AL}		N_{AL}		N_{AL}		N_{AL}	
		12	20	12	20	12	20	12	20
SVM	(rnd)	94.0	94.7	87.5	88.0	91.5	91.9	86.8	87.5
	AL(MCLU)	94.0	95.7	87.8	88.7	91.7	92.2	87.6	87.8
	AL(CBD)	94.3	95.3	87.4	88.4	91.4	92.0	87.6	88.2
	AL(MCLU-ECBD)	94.7	95.9	88.6	89.3	91.9	92.8	88.8	89.0
	AL(MCLU-ABD)	95.3	96.7	88.6	89.3	92.2	93.1	89.1	89.4
	AL(MCLU-KBD)	95.6	96.1	89.6	89.7	92.9	93.4	89.8	89.9
	AL(GBMAL)	95.9	96.5	89.3	89.7	92.6	93.1	89.0	89.5
	CPAL-NCO	**96.9**	**97.1**	**90.6**	**90.9**	**93.5**	**93.7**	**90.5**	**90.7**

AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), and AL(GBMAL), respectively. Random sampling is used as the baseline for the experiments.

For the proposed approach, parameter optimization using grid search is performed over the weights α and β. For AL(MCLU-ABD) and AL(MCLU-KBD), the parameter ρ is optimized using the same approach. For random sampling, the training set $T_R = T_{prop} \cup T_{rnd}$ is employed, where T_{rnd} contains N_{AL} randomly selected instances from U with their corresponding class labels, and T_{prop} is the proper training set. The results for active learning are obtained using the training set $T_{AL} = T_{prop} \cup T_d$, where T_d contains N_{AL} instances selected from U using the aforementioned active learning approaches, with their corresponding class labels. Five trials are conducted to compute the classification accuracy. In each trial, the proper, calibration, training and testing sets are selected at random. For each trial, the best results are selected after parameter optimization and the average classification accuracy is presented.

Figure 6.1 shows the instances selected by the proposed technique for different parameters α and β. It is observed in Fig. 6.1(a) and (g) that when uncertainty is predominant ($\alpha = 0, \beta = 0$), CPAL-NCO selects instances that are concentrated on high uncertainty the regions, *i.e.*, the regions where clusters tend to overlap (near the decision boundaries).

Figure 6.1(f) and (l) shows the instances selected by CPAL-NCO when representativeness is predominant ($\alpha = 0, \beta = 1$). It is observed that the selected instances are located near the cluster centers for the Gaussian database (Fig. 6.1(f)), and they concentrate near coordinate (0,0) for the Two-moon database, which correspond to densely populated regions. On the other hand, when diversity is predominant ($\alpha = 1, \beta = 0$), the instances selected by CPAL-NCO are located in sparsely populated regions. as shown in Fig. 6.1(c) and (i).

Figures 6.1(e) and (k) show the instances selected by CPAL-NCO when uncertainty, diversity, and representativeness are considered together. The parameters for the Gaussian and Two-moon databases are ($\alpha = 1, \beta = 0$) and ($\alpha = 1, \beta = 0$), respectively. It is observed that the selected instances are located in high uncertainty regions, and the spread of the selected instances is lower. In addition, there are no instances located in sparsely populated regions.

Table 6.1 shows the classification accuracy obtained on the synthetic databases for different query functions and values σ, as a function of the number of selected instances N_{AL}. It is observed that the proposed technique outperforms the considered active learning approaches for all the values of σ and N_{AL}. For instance, when the Two-moon database is employed, for $\sigma = 0.08$ and $N_{AL} = 12$, the performance of (rnd), AL(MCLU), AL(MCLU-ABD), AL(MCLU-KBD), AL(CBD), AL(MCLU-ECBD) and AL(GBMAL) is 91.5%, 91.7%, 91.4%, 91.9%, 92.2%, 92.9%, and 92.6%, respectively, whereas that of CPAL-NCO is 93.5%.

To visualize the effect of the parameters α and β on the performance of CPAL-NCO we perform a second experiment. In this experiment, we conduct 100 trials. In each trial, the instances in proper, training, and testing sets are selected at random, along with those in the unlabeled pool, and the average classification accuracy is presented. The proper training set T_{prop} consists of 14 instances, and the number of selected instances from the unlabeled pool is $N_{AL} = 12$.

The classification accuracy for the Gaussian database as a function of α and β for $\sigma = 0.10, 0.12, 0.17,$ and 0.20 is depicted in Fig. 6.2(a), (b), (c), and (d), respectively.

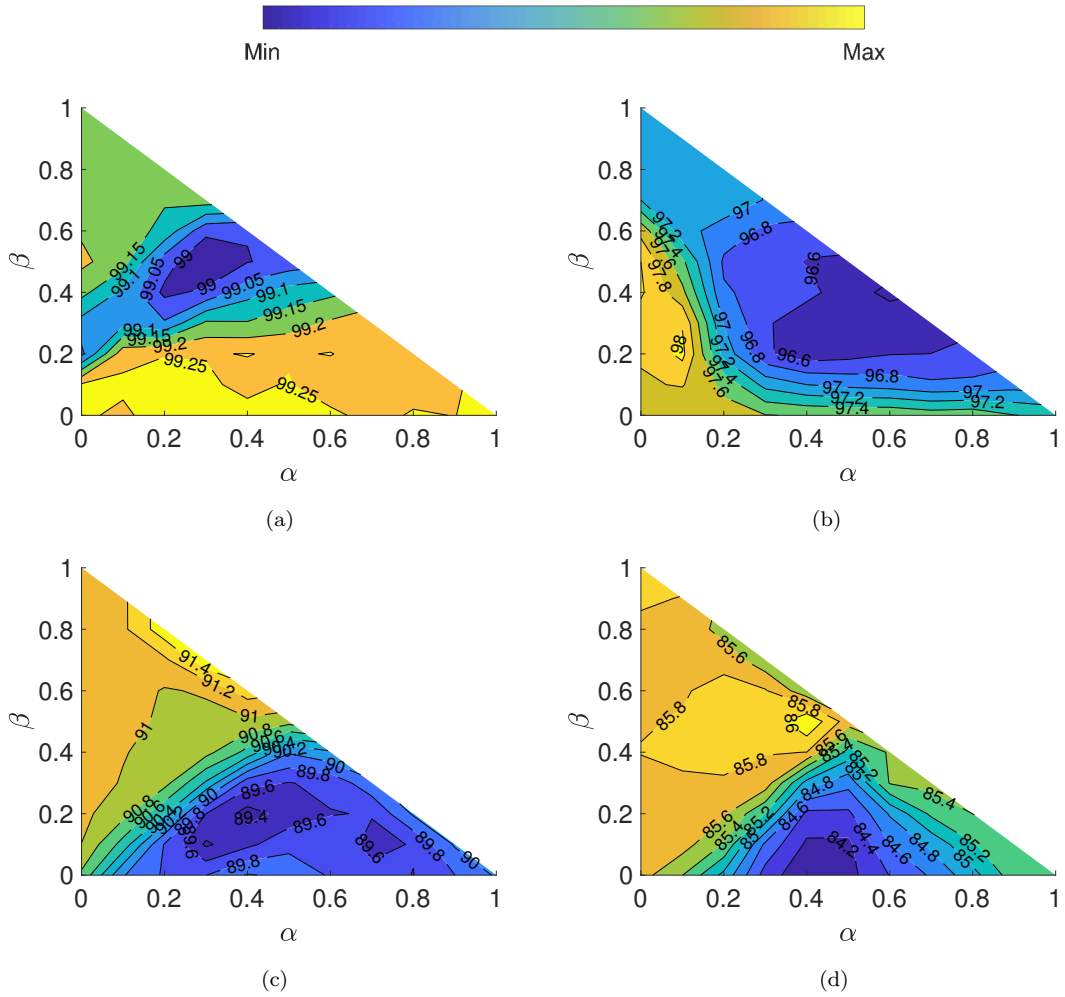


Figure 6.2: Classification accuracy (%) obtained through CPAL-NCO as a function of α and β. Gaussian: (a) $\sigma = 0.10$, (b) $\sigma = 0.12$, (c) $\sigma = 0.17$, (d) $\sigma = 0.20$.

The values of α and β that produce different combinations of uncertainty, diversity, and representativeness are depicted in Fig. 6.4. It is observed that in the low variance (low noise) scenario, *i.e.*, Fig. 6.2(a) and (b), the best performance is obtained for low values of β, and high values of α, which is a combination of uncertainty and diversity (towards the region $(\alpha > 0, \beta = 0)$ in Fig. 6.4). On the other hand, as the variance (noise) increases, Fig. 6.2(c) and (d), it is observed that the parameter β becomes more relevant and the values of α decrease (towards the region $(\alpha = 0, \beta > 0)$ in Fig. 6.4).

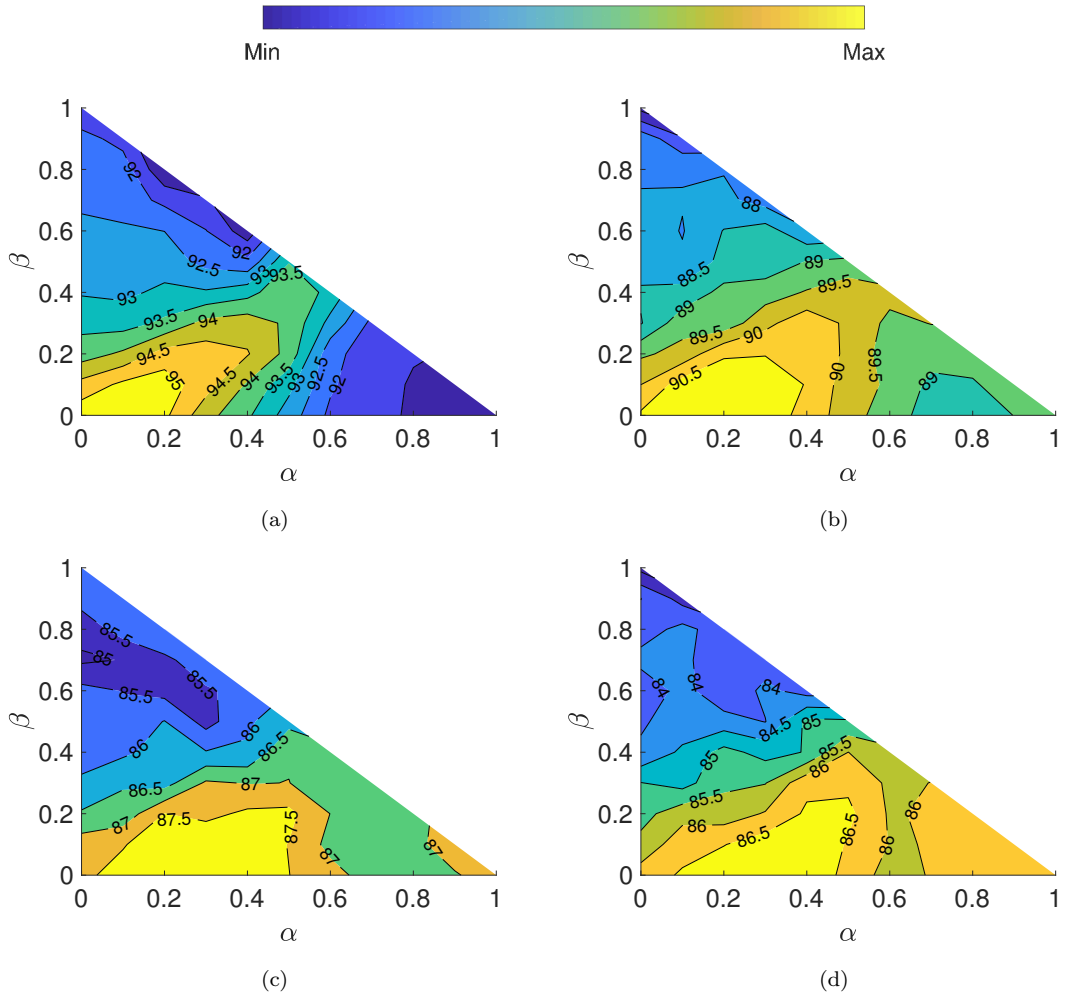


Figure 6.3: Classification accuracy (%) obtained through CPAL-NCO as a function of α and β. Two-moon: (a) $\sigma = 0.08$, (b) $\sigma = 0.13$, (c) $\sigma = 0.16$, (d) $\sigma = 0.18$.

The classification accuracy for the Two-moon database as a function of α and β for $\sigma = 0.08, 0.13, 0.16,$ and 0.18 is shown in Fig. 6.3(a), (b), (c), and (d), respectively. Similar to the results for the Gaussian database, it is observed that for the low noise scenario, Fig. 6.3(a), and (b), high values of α produce the best results (towards the region $(\alpha > 0, \beta = 0)$ in Fig. 6.4). For the high noise case, Fig. 6.3(c), and (d), the parameter β becomes more relevant. Different from the Gaussian database, the parameter β does not need to be increased significantly to produce good performance

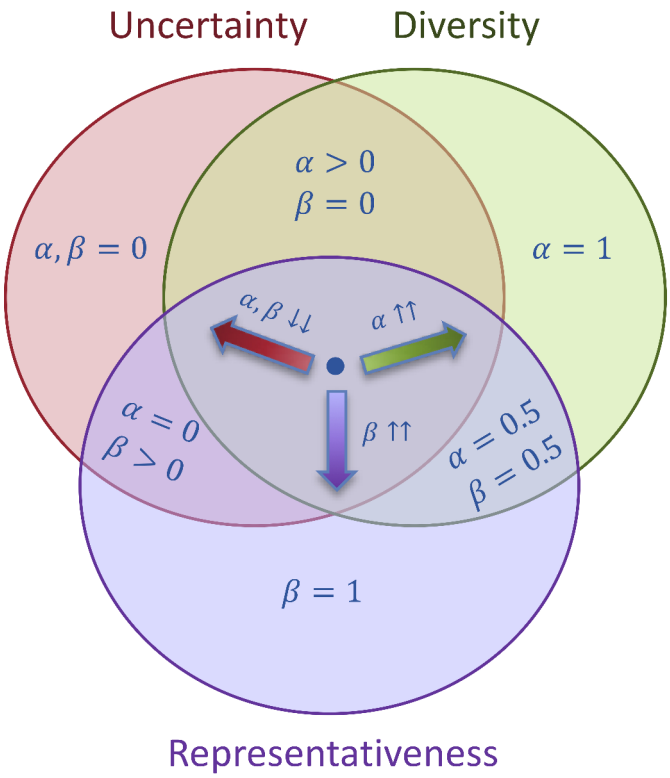


Figure 6.4: Values of α and β that produce different combinations of uncertainty, diversity, and representativeness.

in the high noise scenario for the Two-moon database.

The synthetic database experiments demonstrate that the parameters α and β effectively control the uncertainty, diversity, and representativeness of the selected instances, providing flexibility to the proposed approach. Moreover, it is observed that CPAL-NCO outperforms other existing active learning approaches for classification.

6.5.1.1 Parameter Selection Comparison for CPAL-LR and CPAL-NCO on the Synthetic Databases

Figure. 6.5 shows the performance as a function of parameters η and λ for CPAL-LR, and the performance as a function of α and β for CPAL-NCO, for both the Gaussian and the Two-moon databases. It is observed in Fig. 6.5(a) and (c)

that for the Gaussian database (low-noise, $\sigma = 0.10$) both algorithms perform better using a combination of uncertainty and diversity. On the other hand Fig. 6.5(b) and (d) show that, when the noise is high ($\sigma = 0.20$), a combination of uncertainty and representativeness is a better choice.

Figure. 6.5 (e), (f), (g), and (h) show that the best results for the Two-moon database are obtained for a combination of uncertainty and diversity. However, representativeness becomes more relevant as σ increases.

The results shown in Fig. 6.5 indicate that the trends for the optimal parameters may be similar across different active learning algorithms provided that they consider the same selection criteria.

6.5.1.2 Parameter Selection Modeling for Synthetic Databases

The classification accuracy of CPAL-NCO vs the parameters α and β can be approximated by a surface defined by a fifth order polynomial in two dimensions for ease of use. The polynomial is given by equation (5.6), where we assign $x \leftarrow \alpha$, and $y \leftarrow \beta$. The results of the surface-fitting exercise are shown in Fig. 6.6. It is observed that the surface defined by the fifth order polynomial closely approximates the experimental results. The coefficients of the different polynomials are summarized in Table 6.2.

6.5.2 Face and Object Recognition

Experiments are conducted on two face databases, the Extended YaleB database [63] and the AR face database [64], and one object recognition database, Caltech101 [65][2]. CPAL-NCO is implemented in conjunction with SVMs. We compare the performance improvement obtained through CPAL-NCO with that of the (rnd), AL(MCLU), AL(CBD), AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), and AL(GBMAL).

[2] In this section we use a subset of the Caltech101 database including the following classes: ketch, chandelier, hawksbill, grand piano, brain, butterfly, helicopter, menorah, kangaroo, starfish, trilobit, buddha, ewer, sunflower, scorpion, revolver, laptop, ibis, llama, umbrella, crab, crayfish, cougar face, dragonfly, ferry, flamingo, and lotus.

Table 6.2: Polynomial coefficients for the synthetic databases.

Coefficients	Gaussian		Two-moon	
	$\sigma = 0.10$	$\sigma = 0.20$	$\sigma = 0.08$	$\sigma = 0.18$
p_{00}	95.23	85.75	95.09	83.53
p_{10}	0.16	−2.84	12.61	23.17
p_{01}	1.05	1.10	−5.00	−3.80
p_{20}	−4.85	20.78	−89.55	−126.32
p_{11}	−1.18	21.18	7.19	−25.47
p_{02}	−6.07	−44.46	−9.19	9.93
p_{30}	13.82	−49.39	179.8	278.63
p_{21}	−0.59	−32.23	117.1	134.31
p_{12}	1.16	0.98	−75.14	−50.44
p_{03}	15.38	107.52	24.47	−28.85
p_{40}	−13.24	45.51	−173.30	−274.04
p_{31}	−0.15	14.07	−199.70	−157.52
p_{22}	−4.19	48.28	88.23	34.51
p_{13}	7.91	−67.20	19.82	48.08
p_{04}	−18.14	−84.94	−10.51	35.40
p_{50}	3.99	−13.69	67.35	99.93
p_{41}	2.83	−8.71	75.03	47.85
p_{32}	−4.40	0.88	11.66	23.58
p_{23}	6.52	−23.85	−62.03	−53.52
p_{14}	−7.83	46.32	20.23	1.46
p_{05}	7.84	20.44	−2.80	−2.80

Random sampling is used as the baseline for the experiments, and parameter optimization is performed using exhaustive search, as in the synthetic database experiments.

For each of the experiments in this section, five trials are conducted. In each trial, the proper, calibration, training, and testing sets are selected at random. For each trial, the best results are selected after parameter optimization and the average classification accuracy is presented. The calibration set consists of 199 instances for all the experiments, which results in a resolution of 0.5% in the confidence values calculated, according to equation (2.1). The parameter γ is set to 0.5 in the nonconformity measure given by equation (4.1).

In the following experiments, SVMs are used using the one-vs-all approach. For the Extended YaleB database, the proper training set T_{prop} and U consist of eight and 24 images per class, respectively. The feature descriptors are randomfaces of size

$N = 504$. For the AR database, T_{prop} and U consist of five and 12 images per class, respectively. The feature descriptors are randomfaces of size $N = 540$. For Caltech101, T_{prop} and U consist of ten and 30 images per class, respectively. SIFT descriptors are first extracted. Next, spatial pyramid features, based on the SIFT descriptors, are obtained. The dimension of the spatial pyramid features is then reduced to 3000 through PCA.

6.5.3 Results: CPAL-NCO for Face and Object Recognition

The performance of CPAL-NCO, along with that of the considered active learning approaches, as a function of the number of selected instances N_{AL}, for the different algorithms and databases, is shown in Fig. 6.7, 6.8, and 6.9. It is observed that the performance of the different pattern classification algorithms is significantly improved through active learning, for all the considered databases. Notice that the performance of CPAL-NCO compares favorably with that of the other active learning techniques. This demonstrates the effectiveness of the proposed approach.

The results for the Extended YaleB database in Fig. 6.7 show that the best performance is obtained by CPAL-NCO. Table 6.3 shows that for $N_{AL} = 300$ the classification accuracy of (rnd), AL(MCLU), AL(CBD), AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), AL(GBMAL) is 85.3%, 86.2%, 86.2%, 86.3%, 87.2%, 87.3% and 87.5%, respectively, whereas that of CPAL-NCO is 88.1%.

The results for the AR database in Fig. 6.8. It is observed that the performance of CPAL-NCO is highest among all the considered approaches for the different values of N_{AL}. For instance, for $N_{AL} = 400$, the classification accuracy of (rnd), AL(MCLU), AL(CBD), AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), AL(GBMAL) is 87.1%, 87.2%, 87.6%, 87.9%, 88.2%, 88.4% and 88.7%, respectively, whereas that of CPAL-NCO is 89.5%.

Similar results are obtained for Caltech101, as shown in Fig. 6.9. For instance, when $N_{AL} = 200$, the classification accuracy of (rnd), AL(MCLU), AL(CBD), AL(MCLU-ECBD), AL(MCLU-ABD), AL(MCLU-KBD), AL(GBMAL) is 70.5%, 71.0%,

Table 6.3: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}.

Algorithm	Query function	YaleB			AR			Caltech101		
		300	400	500	300	400	500	200	300	400
SVM	(rnd)	85.3	88.0	89.6	85.2	87.1	89.0	70.5	71.9	73.2
	AL(MCLU)	86.2	88.2	89.9	84.9	87.2	89.1	71.0	74.2	74.6
	AL(CBD)	86.2	88.4	90.1	85.5	87.6	90.2	71.2	74.2	74.6
	AL(MCLU-ECBD)	86.3	88.5	90.0	85.8	87.9	90.5	71.4	74.5	74.8
	AL(MCLU-ABD)	87.2	88.6	90.4	86.2	88.2	90.4	72.8	74.9	76.6
	AL(MCLU-KBD)	87.3	89.1	90.5	86.1	88.4	90.3	71.9	75.0	76.7
	AL(GBMAL)	87.5	89.2	90.7	86.3	88.7	90.9	72.5	75.0	76.6
	CPAL-NCO	**88.1**	**89.7**	**91.4**	**87.1**	**89.5**	**91.4**	**74.3**	**75.9**	**77.4**

71.2%, 71.4%, 72.8%, 71.9% and 72.5%, respectively, whereas that of CPAL-NCO is 74.3%.

Figure 6.10 shows the classification accuracy of CPAL-NCO as a function of the parameters α and β for the Extended YaleB, AR, and Caltech101 databases (average over the five trials for the different values of α and β). It is observed that the best performance is obtained for a combination of uncertainty, diversity, and representativeness, *i.e.*, $\alpha, \beta \geq 0$, for all the considered databases. For the Extended YaleB database ($N_{AL} = 600$), the best performance is obtained in the region $\alpha \in [0.4, 0.7]$ and $\beta \in [0, 0.3]$, and the classification accuracy peaks when $\alpha = 0.6$ and $\beta = 0.1$ (91.6%). For the AR database ($N_{AL} = 100$), the best performance is obtained in the region $\alpha \in [0.2, 0.4]$ and $\beta \in [0.2, 0.4]$, and the classification accuracy peaks when $\alpha = 0.4$ and $\beta = 0.3$ (79.3%). For Caltech101 ($N_{AL} = 500$), the best performance is obtained in the region $\alpha \in [0, 0.4]$ and $\beta \in [0.4, 0.8]$, and the classification accuracy peaks when $\alpha = 0.1$ and $\beta = 0.6$ (73.4%).

The results on face and object recognition show that CPAL-NCO improves the performance of several pattern classification algorithms across different databases, outperforming other state-of-the-art active learning techniques.

The confusion matrix for ten classes (trilobite, buddha, ewer, sunflower, scorpion, revolver, laptop, ibis, llama, and umbrella), for two different sets of parameters

α and β, when SVMs are used, is shown in Fig. 6.11 ($\alpha = 0$, $\beta = 0.6$) and Fig. 6.12 ($\alpha = 0.4$, $\beta = 0.5$). It is observed that for ($\alpha = 0.4$, $\beta = 0.5$) the performance is better (85.9%), compared to that obtained for ($\alpha = 0$, $\beta = 0.6$) (83.7%).

Figure 6.11 shows that three elements of class buddha are classified as ewer, when ($\alpha = 0$, $\beta = 0.6$). On the other hand, when ($\alpha = 0.4$, $\beta = 0.5$), only one element of class buddha is regarded as ewer. Fig. 6.13(a), (b), and (c) show the images of class buddha that are regarded as ewer when ($\alpha = 0$, $\beta = 0.6$), while only Fig. 6.13(c) is regarded as ewer when ($\alpha = 0.4$, $\beta = 0.5$). This shows the importance of parameter selection in the proposed technique. An example image of class ewer is shown in Fig. 6.13(d).

Moreover, Fig. 6.11 shows that three elements of class ibis are classified as llama, when ($\alpha = 0$, $\beta = 0.6$). On the other hand, when ($\alpha = 0.4$, $\beta = 0.5$), none of the elements of class buddha are regarded as ewer. Fig. 6.13(e), (f), and (g) show the images of class ibis that are regarded as llama when ($\alpha = 0$, $\beta = 0.6$). An example image of class llama is shown in Fig. 6.13(h).

6.5.4 Applications to Video for Emotion Recognition

In the following experiments we employ the Oulu-CASIA NIR&VIS database. The proposed approach is shown in Fig. 6.14. For each one of the video snippets, optical flow is obtained for the different frames using the technique described in [73]. Optical flow is then converted into RGB images, and averaged across the video frames to mitigate noise. The resulting image is normalized, dividing by the highest optical flow magnitude across the video snippet, then cropped to 200×200 pixels and rescaled to 50×50 pixels. For the following experiments we use the CNN described in Table 6.4.

For each of the experiments in this section, 5 trials are conducted. In each trial, the order of the instances in the training set is permuted. The average classification accuracy is presented. The proper training set T_{prop} consists of 120 images per class, and the unlabeled pool U consists of 70 images per class. The calibration set consists

Table 6.4: CNN architecture for the Oulu-CASIA database.

Layers	Filter Size	Stride	Padding	Output $W \times H \times L$
Input	-	-	-	$50 \times 50 \times 3$
Conv-ReLU Avg_pool	7×7 2×2	1 2	0	$22 \times 22 \times 20$
Conv-ReLU Avg_pool	7×7 2×2	1 2	0	$8 \times 8 \times 60$
FC-ReLU Dropout	-	-	-	500
FC-Softmax	-	-	-	6

Table 6.5: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL}.

Query function	Oulu-CASIA NIR&VIS		
	No. instances N_{AL}		
	50	150	200
(rnd)	61.1	64.2	66.5
AL(MCLU)	63.2	64.7	68.0
AL(MCLU-ABD)	64.2	67.2	69.2
AL(MCLU-KBD)	64.9	66.8	68.6
AL(GBMAL)	63.7	67.3	68.6
CPAL-NCO	**66.2**	**68.1**	**69.5**

of 99 instances, and the parameter γ is set to 0.5 in the nonconformity measure given by equation (4.1).

The classification accuracy for different values of N_{AL} can be seen in Fig 6.15. Notice that the performance of CPAL-NCO compares favorably with that of the other active learning techniques. The results are summarized in Table 6.5. It is observed that for $N_{AL} = 100$ the classification accuracy of (rnd), AL(MCLU), AL(MCLU-ABD), AL(MCLU-KBD), AL(GBMAL) is 64.2%, 64.7%, 67.2%, 66.8%, and 67.3%, respectively, whereas that of CPAL-NCO is 68.1%. Similar results are obtained for $N_{AL} = 50$ and $N_{AL} = 300$.

Table 6.6: Execution time of different active learning techniques as a function of the number of selected instances N_{AL} (AR database).

Query function	Execution time (s)		
	No. instances N_{AL}		
	200	300	400
(rnd)	2.0×10^{-4}	2.2×10^{-4}	2.4×10^{-4}
AL(MCLU)	0.9	0.9	0.9
AL(MCLU-ABD)	1.9	3.1	3.9
AL(MCLU-KBD)	2.1	3.7	5.0
CPAL-CNN	2.4	4.0	5.2
CPAL-LR	1.3	1.3	1.4
CPAL-NCO	11.9	11.7	12.0

6.6 Execution Time of Active Learning Approaches

We calculate the average execution time of (rnd), AL(MCLU), AL(MCLU-ABD), AL(MCLU-KBD), CPAL-CNN, CPAL-LR, and CPAL-NCO. The execution times are averaged over 10 iterations, and the results are summarized in Table 6.6. The algorithms are tested using MATLAB running on a 2.5 GHz Intel Core i7 processor. It is observed that (rnd) is the simplest approach, with execution times in the order of 10^{-4}. However, its performance is the lowest, as shown in Sections 3, 4, 5, and 6. Notice that the execution times of CPAL-LR (≈ 1.3) and CPAL-NCO (≈ 11.9) remain similar for the different values of N_{AL}, since these two approaches compute the relevance for all the instances in the unlabeled pool, regardless of the value of N_{AL}. The execution times of AL(MCLU-ABD), AL(MCLU-KBD), and CPAL-CNN are similar, and increase with the number of selected instances N_{AL}.

6.7 Chapter Conclusion

A conformal prediction based active learning algorithm is presented in this chapter. The proposed approach uses a novel query function that determines the relevance of unlabeled instances through the solution of a nonlinear constrained optimization problem. The proposed query function considers uncertainty, diversity, and representativeness for instance selection. Moreover, experiments performed on synthetic data

provide a greater insight into the working mechanism of the proposed approach and also show the behavior of the parameters α and β for two different synthetic databases.

CPAL-NCO is implemented in conjunction with SVMs and CNNs. Experiments conducted on face, object, and emotion recognition databases show that CPAL-NCO outperforms previous work on active learning, improving performance across different pattern classification techniques and databases.

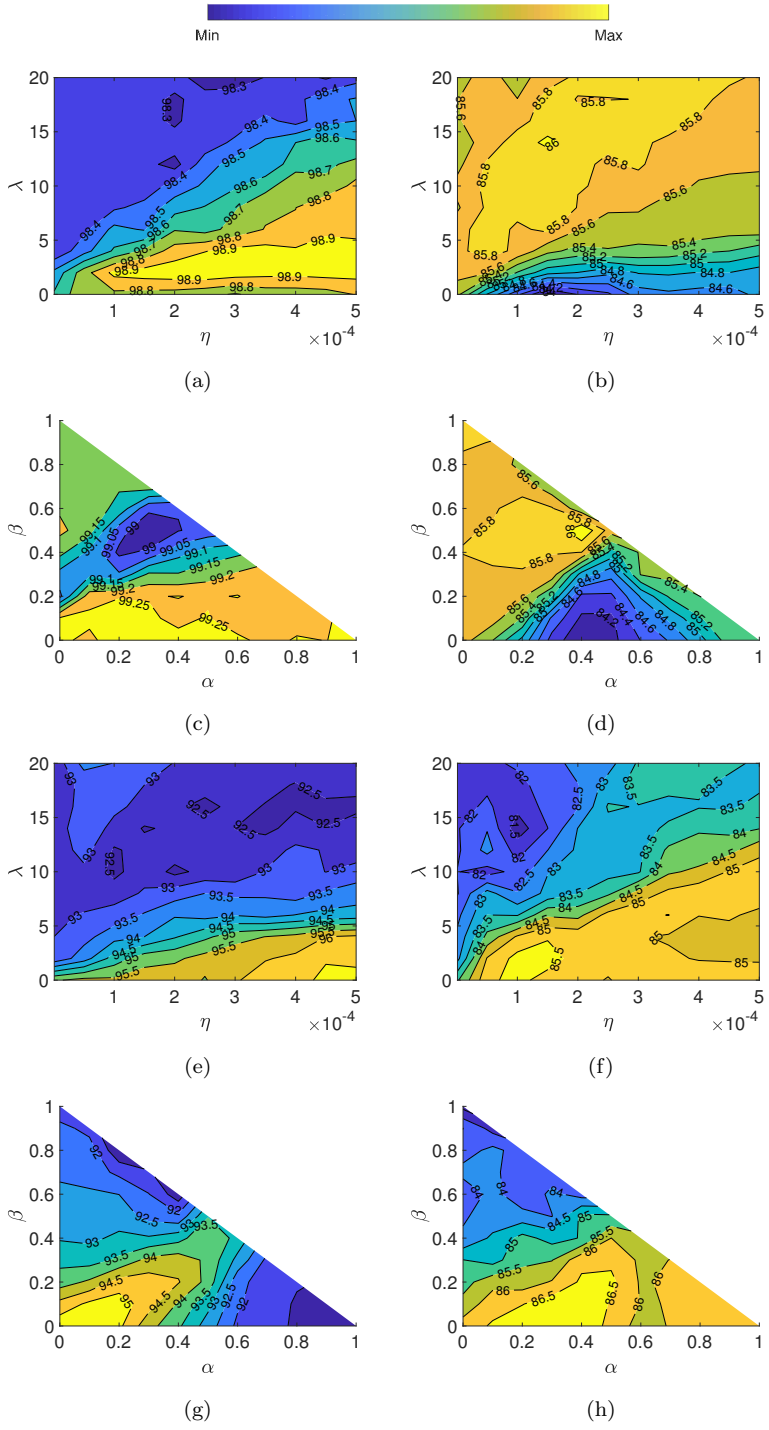


Figure 6.5: Classification accuracy (%) obtained through CPAL-LR, and CPAL-NCO as a function of η, λ, α, and β. Gauss: (a) $\sigma = 0.10$, (CPAL-LR) (b) $\sigma = 0.20$ (CPAL-LR), (c) $\sigma = 0.10$ (CPAL-NCO), (d) $\sigma = 0.20$ (CPAL-LR). Two-moon: (e) $\sigma = 0.08$ (CPAL-LR), (f) $\sigma = 0.18$ (CPAL-LR), (g) $\sigma = 0.08$ (CPAL-NCO), (h) $\sigma = 0.18$ (CPAL-LR).

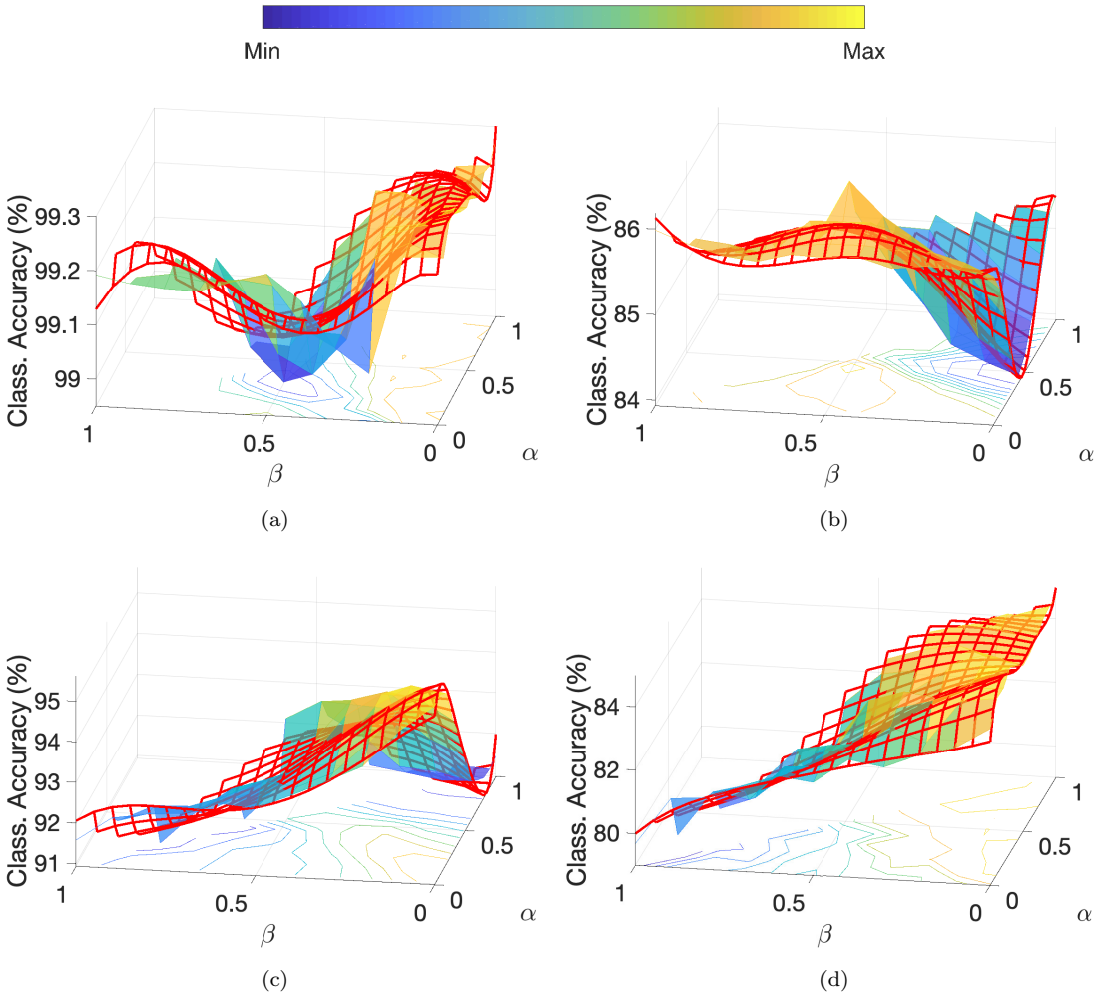


Figure 6.6: Characterization of the surface defined by performance vs parameters α and β using a fifth order polynomial in two dimensions (fitted surface shown in red). Gaussian: (a) $\sigma = 0.10$, (b) $\sigma = 0.20$, Two-moon: (c) $\sigma = 0.08$, (d) $\sigma = 0.18$.

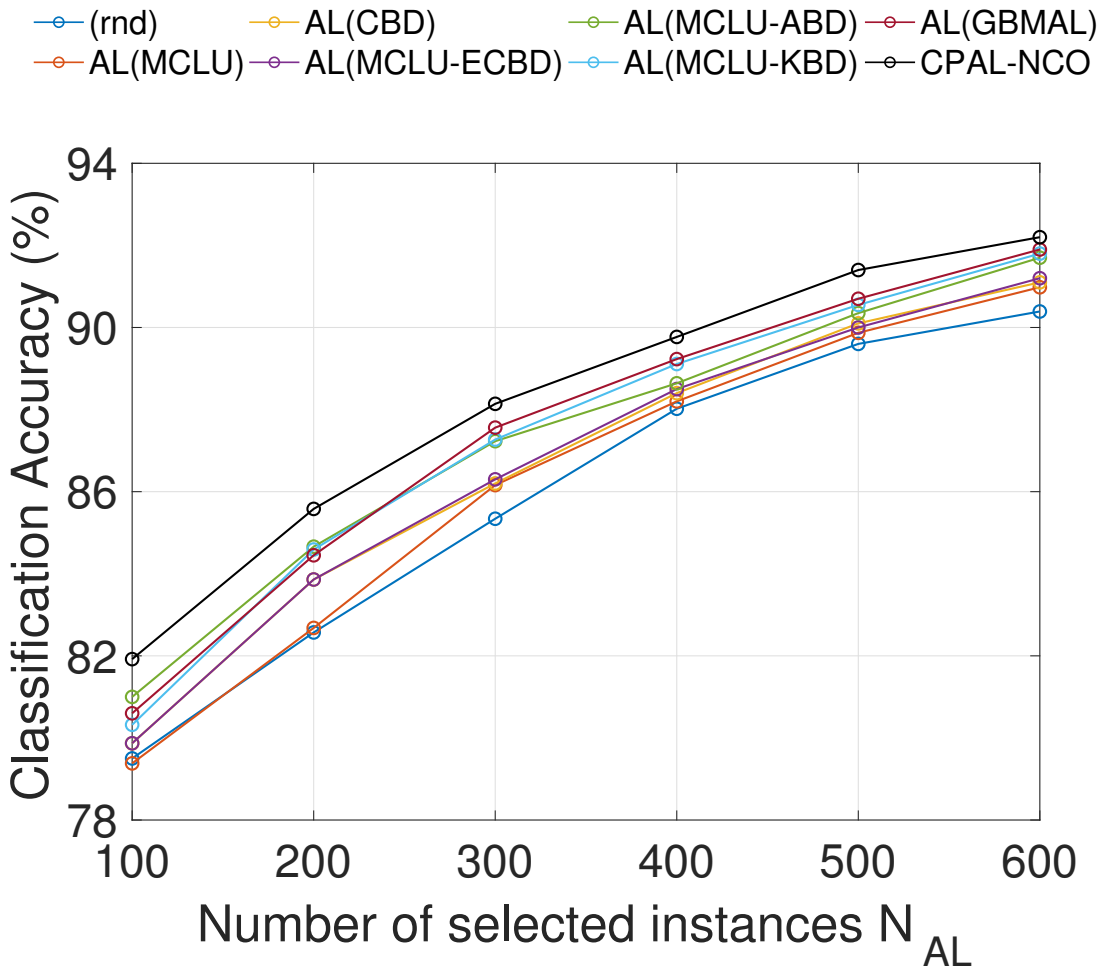


Figure 6.7: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL} (YaleB).

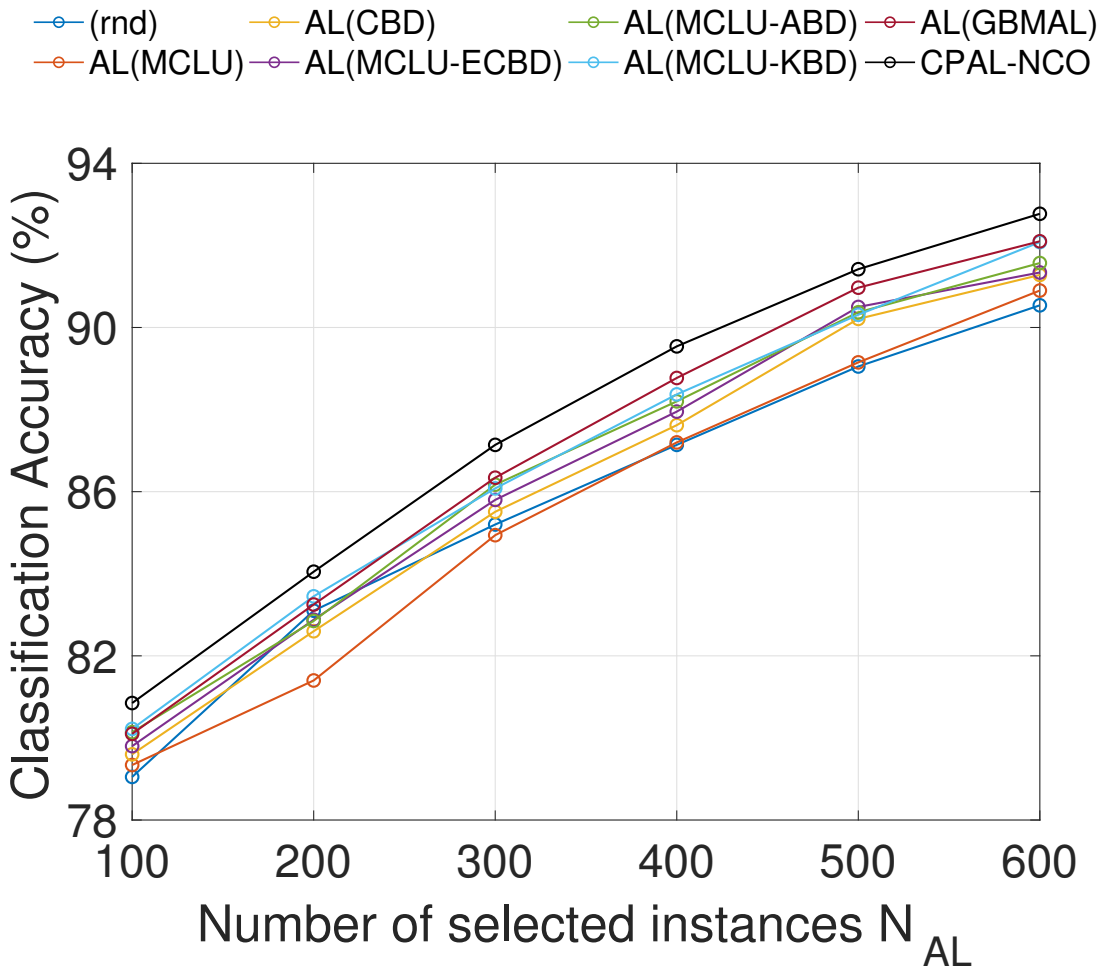


Figure 6.8: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL} (AR).

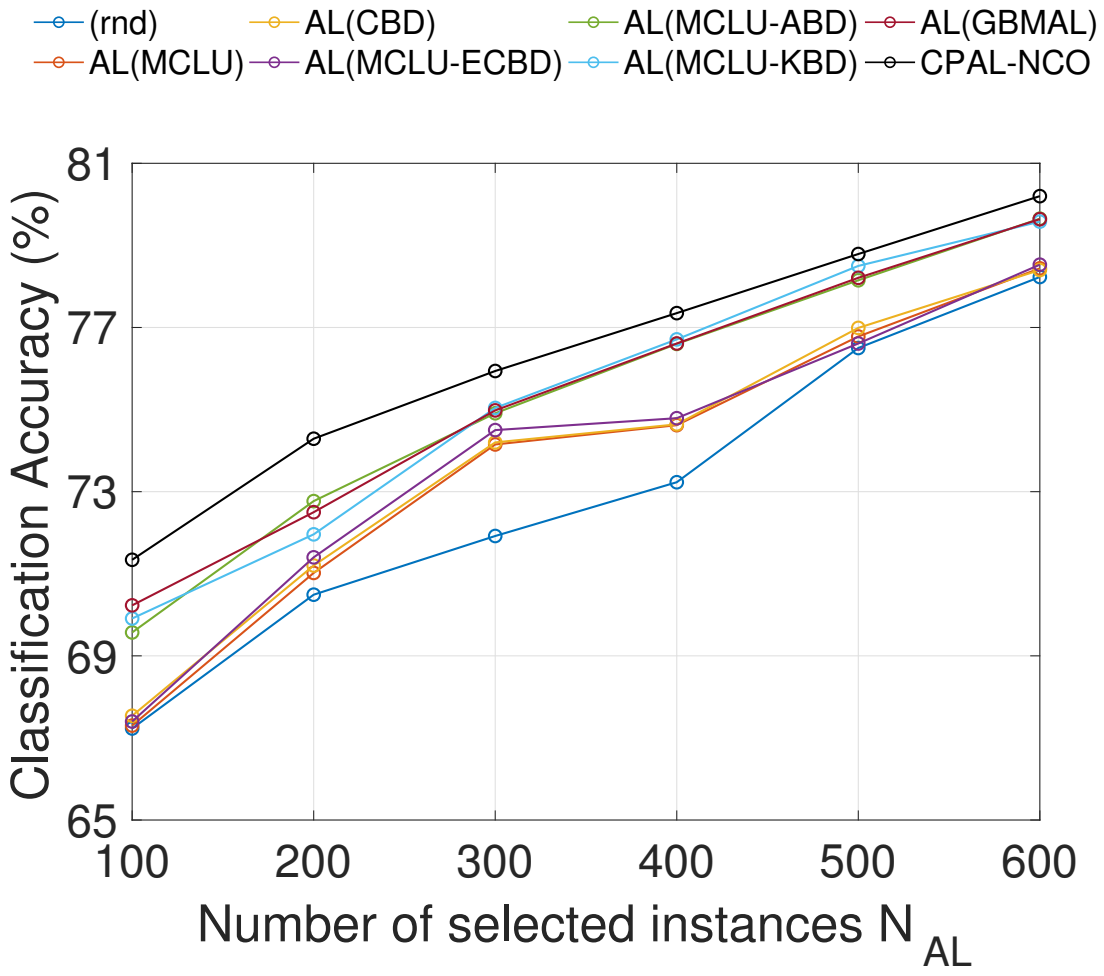


Figure 6.9: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL} (Caltech101).

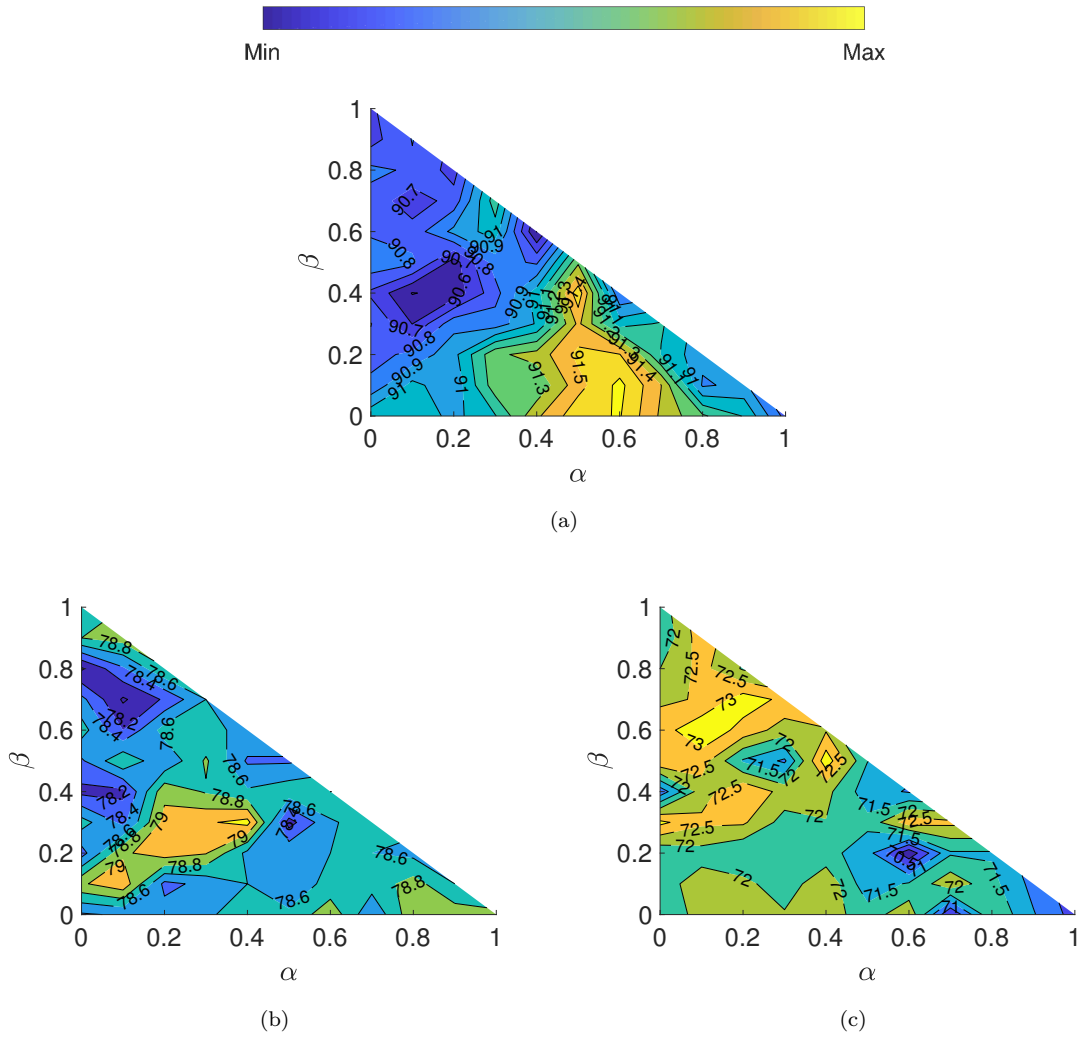


Figure 6.10: Classification accuracy (%) obtained through CPAL-NCO (SVMs) as a function of α and β: (a) YaleB ($N_{AL} = 600$), (b) AR ($N_{AL} = 100$), (c) Caltech101 ($N_{AL} = 500$).

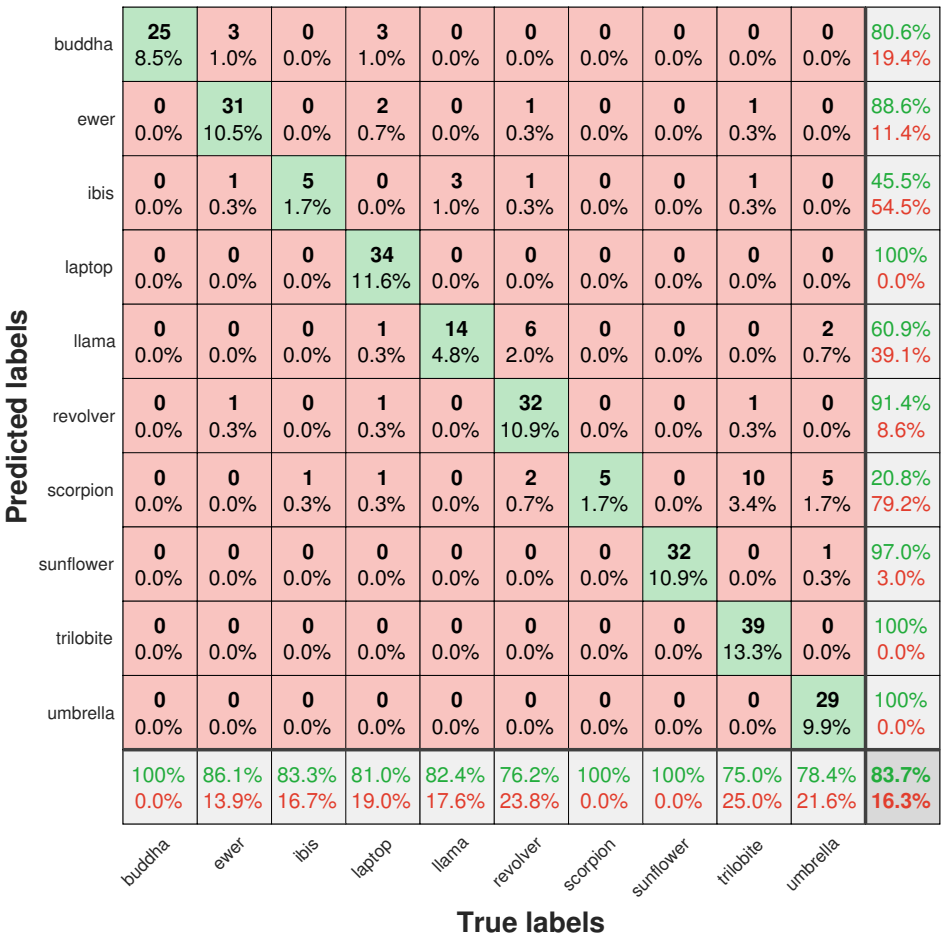


Figure 6.11: Confusion matrix for classes trilobite, buddha, ewer, sunflower, scorpion, revolver, laptop, ibis, llama, and umbrella ($\alpha = 0$, $\beta = 0.6$).

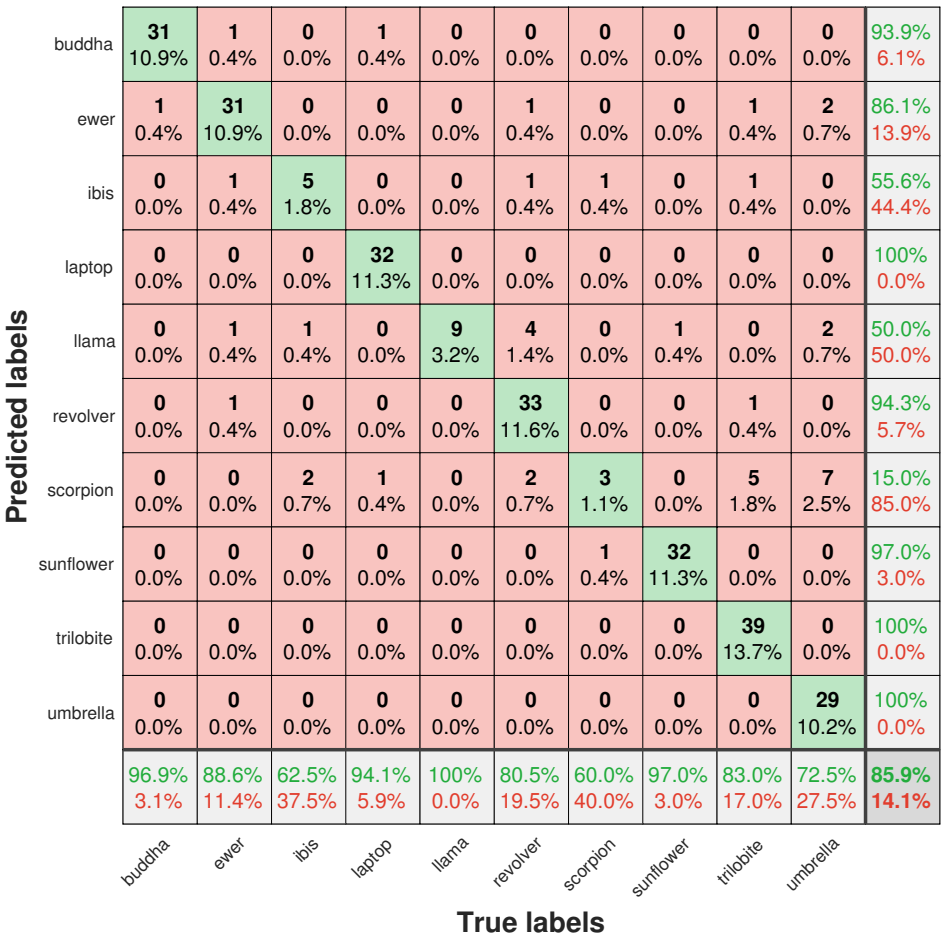
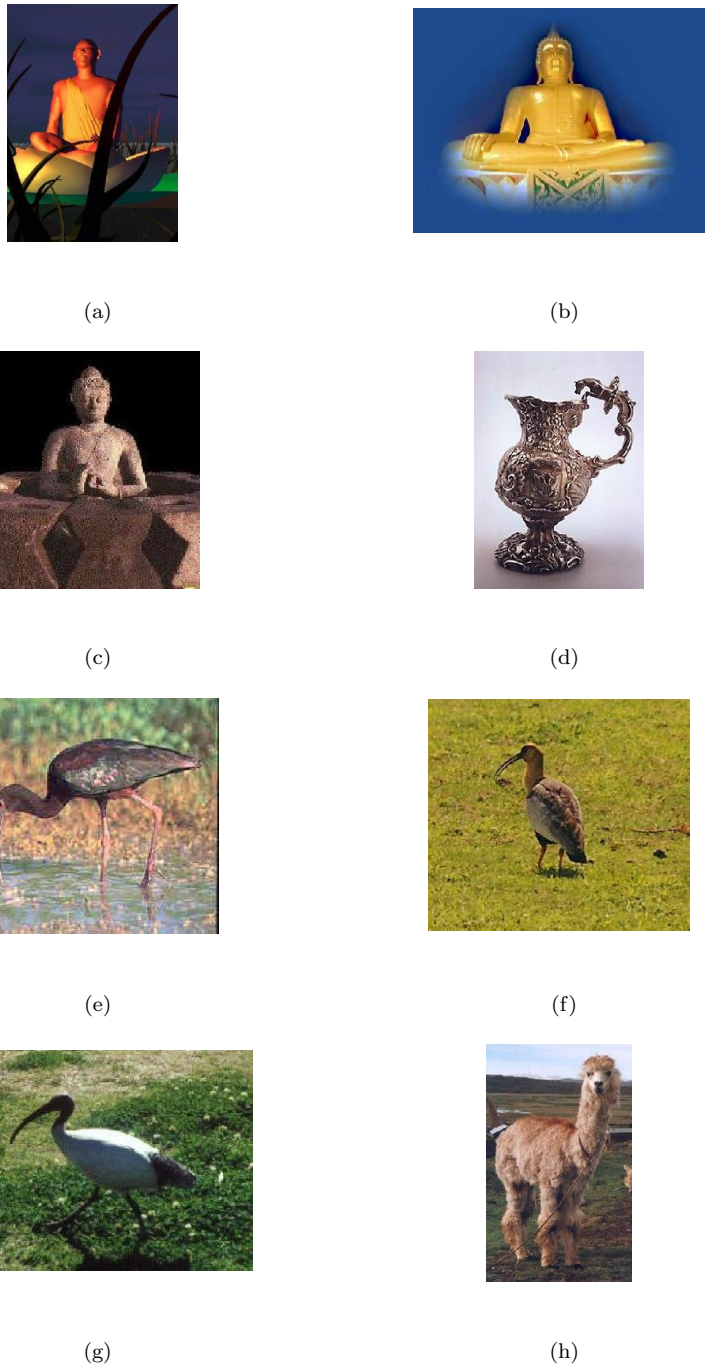


Figure 6.12: Confusion matrix for classes trilobite, buddha, ewer, sunflower, scorpion, revolver, laptop, ibis, llama, and umbrella ($\alpha = 0.4$, $\beta = 0.5$).



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 6.13: Caltech101 images of class buddha that are regarded as ewer by filename: (a) image_0007.jpg, (b) image_0045.jpg, and (c) image_0046.jpg. Images of class ibis that are regarded as llama: (e) image_0056.jpg, (f) image_0022.jpg, and (g) image_0028.jpg. Example image of class ewer (d) image_0010.jpg, and llama (h) image_0001.jpg.

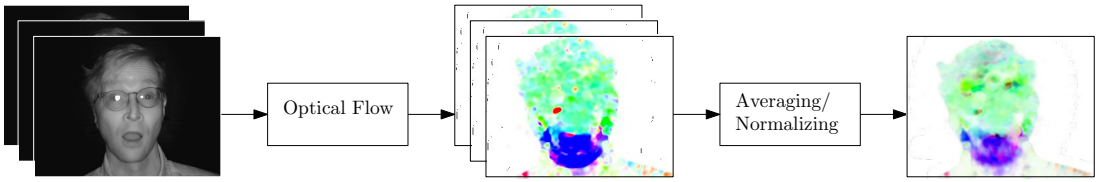


Figure 6.14: Feature extraction using optical flow.

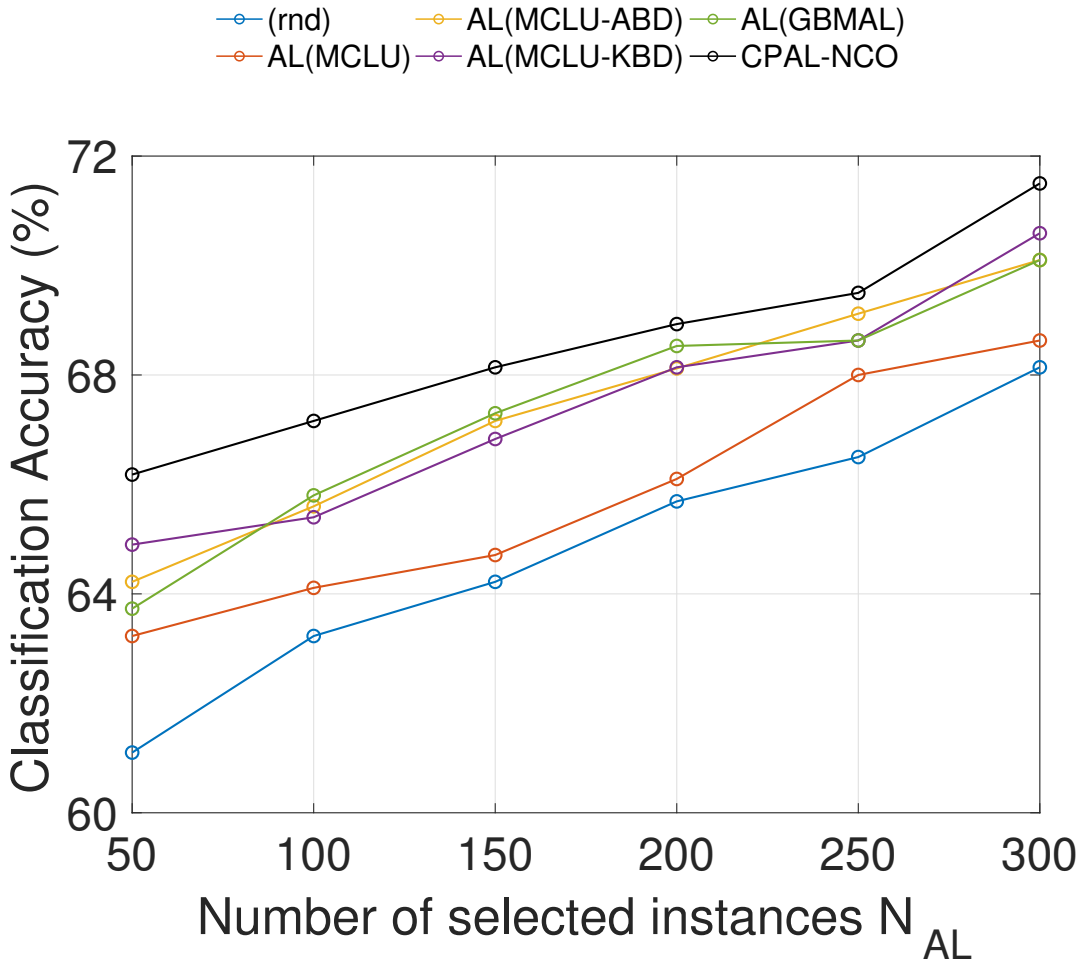


Figure 6.15: Classification accuracy (%) using different active learning techniques as a function of the number of selected instances N_{AL} for the Oulu-CASIA database.

Chapter 7

CONCLUSIONS AND FUTURE WORK

Several conformal prediction based active learning approaches are proposed in this work. The proposed techniques use novel query functions that determine the relevance of unlabeled instances considering uncertainty, diversity, and representativeness. Moreover, several nonconformity measures that produce reliable confidence values are presented. The contributions made in this work are categorized in Fig. 7.1 [74, 75, 76, 77], along with previous work on batch mode active learning.

The proposed techniques are implemented in conjunction with three different pattern classification algorithms: SVMs, sparse coding, and CNNs, outperforming previous work on active learning. Experiments are conducted on two face recognition databases, Extended YaleB and AR, one object recognition database, Caltech101, and one emotion recognition video database, Oulu-CASIA NIR&VIS. The experimental results demonstrate the improved performance obtained through the proposed techniques.

In addition to performance enhancement, conformal prediction based active learning produces reliable confidence values that are used to predict class labels with guaranteed error rate. The quality of the confidence values is demonstrated experimentally using three different metrics: the evaluation of the validity property, the percentage of singleton predictions, and the average number of class labels in the prediction sets.

As part of future work, more efficient ways to compute information density can be explored, since the kernel method and k-nearest neighbors approach used in this work become computationally intensive for large databases. Recent work has

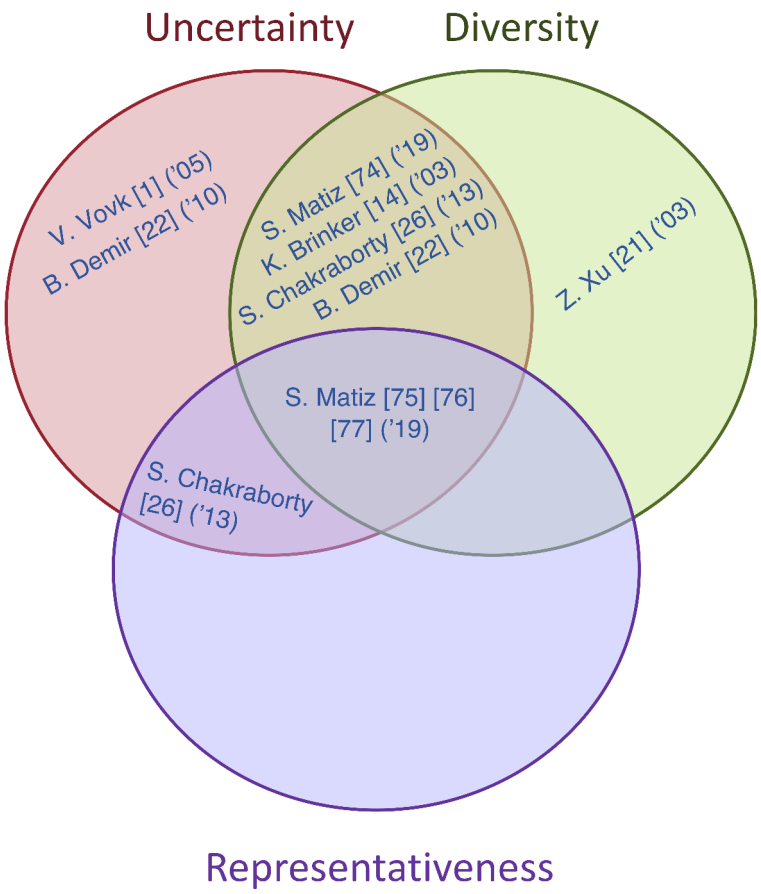


Figure 7.1: Contributions and related work on batch mode active learning.

addressed this issue using density estimation based on mass [78], which we plan to investigate on our ongoing work. Moreover, when the proper training set is small, supervised DML algorithms, such as LMNN [37], may overfit the data, degrading the performance of the subsequent active learning stage. Therefore, further improvements may be obtained by investigating semi-supervised DML techniques [79, 80], which exploit the information of both labeled and unlabeled data. Future developments of this work also include parameter estimation using optimization algorithms to obtain the query function weights, thereby avoiding grid search procedures.

BIBLIOGRAPHY

[1] V. Vovk, A. Gammerman, and G. Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[2] A. Gammerman, I. Nouretdinov, B. Burford, A. Chervonenkis, V. Vovk, and L. Zhiyuan. Clinical mass spectrometry proteomic diagnosis by conformal predictors. *Statistical applications in genetics and molecular biology*, 7(2), 2008.

[3] I. Nouretdinov, S. G. Costafreda, A. Gammerman, A. Chervonenkis, V. Vovk, V. Vapnika, and C. H. Fu. Machine learning classification with confidence: application of transductive conformal predictors to mri-based diagnostic and prognostic markers in depression. *Neuroimage*, 56(2):809–813, 2011.

[4] Y. Luo, Bsoul, A. A. R. Besoul, and K. Najarian. Confidence based classification with dynamic conformal prediction and its applications in biomedicine. In *Proc. in Engineering in Medicine and Biology Society (EMBC)*, 2011.

[5] V. Balasubramanian, S. S. Ho, and V. Vovk. *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. Newnes, 2014.

[6] L. Valiant. *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books, 2013.

[7] I. Nouretdinov, V. Vovk, M. Vyugin, A. Gammerman, and P. Perona. Pattern recognition and density estimation under the general iid assumption. In *Proc. 14th Annual Conference on Computational Learning Theory (COLT'01) and 5th European Conference on Computational Learning Theory (EuroCOLT'01)*, pages 337–353. Springer, 2001.

[8] T. Melluish, S. Craig, I. Nouretdinov, and V. Vovk. Comparing the bayes and typicalness frameworks. In *Proc. European Conference on Machine Learning (ECML 2001)*, volume 2167, pages 360–371. Springer, 2009.

[9] S. S. Ho and H. Wechsler. Query by transduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(9):1557–1571, Sep. 2008.

[10] V. Balasubramanian, S. Chakraborty, and S. Panchanathan. Generalized query by transduction for online active learning. *Proc. Int. Conf. Computer Vision Workshops (ICCV Workshops)*, pages 1378–1385, 2009.

[11] H. Papadopoulos, H. Boström, and T. Löfström. Conformal prediction using decidion trees. *Proc. IEEE Int. Conf. Data Mining (ICDM)*, pages 330–339, Dec. 2013.

[12] H. Papadopoulos, V. Vovk, and A. Gammerman. Conformal prediction with neural networks. *Proc. IEEE Int. Conf. Tools with Artificial Intelligence (ICTAI)*, 2:388–395, Oct. 2007.

[13] U. Johansson T. Löfström and H. Boström. Effective utilization of data in inductive conformal prediction using ensembles of neural networks. *Proc. The International Joint Conference on Neural Networks (IJCNN)*, 2:1–8, Aug. 2013.

[14] K. Brinker. Incorporating diversity in active learning with support vector machines. In *Proc. Int. Conf. Machine Learning (ICML)*, 2003.

[15] J. Li, J. M. Bioucas-Dias, and A. Plaza. Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4085–4098, 2010.

[16] Q. Shi, B. Du, and L. Zhang. Spatial coherence-based batch-mode active learning for remote sensing image classification. *IEEE Trans. Image Process.*, 24(7):2037–2050, 2015.

[17] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *European Conference on Information Retrieval*, pages 246–257. Springer, 2007.

[18] S. S. Ho and H. Wechsler. Query by transduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(9):1557–1571, Sept. 2008.

[19] C. Monteleoni and M. Kaariainen. Practical online active learning for classification. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[20] D. Sculley. Online active learning methods for fast label-efficient spam filtering. In *CEAS*, volume 7, page 143, 2007.

[21] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. In *Proc. in European Conference on Information Retrieval*, 2003.

[22] B. Demir, C. Persello, and L. Bruzzone. Batch-mode active-learning methods for the interactive classification of remote sensing images. *IEEE Trans. Geosci. Remote Sens*, 49(3):1014–1031, Oct. 2010.

[23] R. Wang and S. Kwong. Active learning with multi-criteria decision making systems. *Pattern Recognition*, 47(9):3106–3119, 2014.

[24] S. Chakraborty, V. Balasubramanian, and S. Panchanathan. Adaptive batch mode active learning. *IEEE Trans. Neural Netw. Learn. Syst.*, 26(8):1747–1760, 2015.

[25] J. Sourati, M. Akcakaya, D. Erdogmus, T. K. Leen, and J. G. Dy. A probabilistic active learning algorithm based on fisher information ratio. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(8):2023–2029, 2018.

[26] S. Chakraborty, V. Balasubramanian, and S. Panchanathan. Generalized batch mode active learning for face-based biometric recognition. *Pattern Recognition*, 46(2):497–508, 2013.

[27] S. Chakraborty, V. Balasubramanian, Q. Sun, S. Panchanathan J., and Ye. Active batch selection via convex relaxations with guaranteed solution bounds. *IEEE transactions on pattern analysis and machine intelligence*, 37(10):1945–1958, 2015.

[28] Y. Gu, Z. Jin, and S. C. Chiu. Active learning combining uncertainty and diversity for multi-class image classification. *IET Computer Vision*, 9(3):400–407, 2014.

[29] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, pages 1070–1079, 2008.

[30] X. Li and Y. Guo. Adaptive active learning for image classification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 859–866, 2013.

[31] Q. Li, X. Shi, L. Zhou, Z. Bao, and Z. Guo. Active learning via local structure reconstruction. *Pattern Recognition Letters*, 92:81–88, 2017.

[32] Z. Wang, B. Du, L. Zhang, L. Zhang, and X. Jia. A novel semisupervised active-learning algorithm for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(6):3071–3083, 2017.

[33] B. Du, Z. Wang, L. Zhang, L. Zhang, and D. Tao. Robust and discriminative labeling for multi-label active learning based on maximum correntropy criterion. *IEEE Transactions on Image Processing*, 26(4):1694–1707, 2017.

[34] G. Wang, J. N. Hwang, C. Rose, and F. Wallace. Uncertainty-based active learning via sparse modeling for image classification. *IEEE Transactions on Image Processing*, 28(1):316–329, 2019.

[35] S. Kee, E. del Castillo, and G. Runger. Query-by-committee improvement with diversity and density in batch active learning. *Information Sciences*, 454:401–418, 2018.

[36] E. P. Xing, M. I Jordan, S. J. Russell, and A. Y. Ng. Distance metric learning with application to clustering with side-information. In *Proc. Advances in neural information processing systems*, pages 521–528, 2003.

[37] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

[38] J. Wang, Z. Deng, K. S. Choi, Y. Jiang, X. Luo, F. L. Chung, and S. Wang. Distance metric learning for soft subspace clustering in composite kernel space. *Pattern Recognition*, 52:113–134, 2016.

[39] B. Nguyen, C. Morell, and B. De Baets. Distance metric learning with the universum. *Pattern Recognition Letters*, 100:37–43, 2017.

[40] S. Chakraborty and S. Das. k- means clustering with a new divergence-based distance metric: Convergence and performance analysis. *Pattern Recognition Letters*, 100:67–73, 2017.

[41] O. Chapelle, P. Haffner, and V. N. Vapnik. Support vector machines for histogram-based image classification. *IEEE transactions on Neural Networks*, 10(5):1055–1064, 1999.

[42] H. Drucker, D. Wu, and V. N. Vapnik. Support vector machines for spam categorization. *IEEE Transactions on Neural networks*, 10(5):1048–1054, 1999.

[43] Z. Jiang, Z. Lin, and L. S. Davis. Label consistent K-SVD: Learning a discriminative dictionary for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2651–2664, Nov 2013.

[44] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Trans. Image Process.*, 17(1):53–69, Jan 2008.

[45] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1794–1801, 2009.

[46] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[47] M. Yang, H. Chang, and W. Luo. Discriminative analysis-synthesis dictionary learning for image classification. *Neurocomputing*, 219:404–411, 2017.

[48] L. Zhang, L. Zhang, D. Tao, and X. Huang. Tensor discriminative locality alignment for hyperspectral image spectral–spatial feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 51(1):242–256, 2013.

[49] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du. Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding. *Pattern Recognition*, 48(10):3102–3112, 2015.

[50] Y. Dong, B. Du, and L. Zhang. Target detection based on random forest metric learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(4):1830–1838, 2015.

[51] S. Matiz and K. E. Barner. Label consistent recursive least squares dictionary learning for image classification. *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pages 1888–1892, 2016.

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[53] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. Cost-effective active learning for deep image classification. *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[54] C. Käding, E. Rodner, A. Freytag, and J. Denzler. Active and continuous exploration with deep neural networks and expected model output changes. *arXiv preprint arXiv:1612.06129*, 2016.

[55] S. Otálora, O. Perdomo, F. González, and H. Müller. Training deep convolutional neural networks with active learning for exudate classification in eye fundus images. In *Intravascular Imaging and Computer Assisted Stenting, and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis*, pages 146–154. Springer, 2017.

[56] G. Shafer and V. Vock. A tutorial on conformal prediction. *J. Mach. Learn. Res. 9*, pages 371–421, 2008.

[57] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Proc. Int. Conf. Machine Learning (ICML)*, 2000.

[58] G. Schohn and D. Cohn. Less is more: Active learning with support vector machines. In *Proc. Int. Conf. Machine Learning (ICML)*, 2000.

[59] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang. Representative sampling for text classification using support vector machines. In *European Conference on Information Retrieval*, pages 393–407. Springer, 2003.

[60] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, 2003.

[61] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *ACM Sigmod Record*, volume 29, pages 427–438. ACM, 2000.

[62] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6(Jun):937–965, 2005.

[63] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[64] A. M. Martinez and R. Benavente. The AR face database. *CVC Tech. Report # 24*, 1998.

[65] L. FeiFei, R. Fergus, and P. Perona. Learning generative visual models from few training samples: An incremental bayesian approach tested on 101 object categories. *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, 2004.

[66] G. Zhao, X. Huang, M. Taini, S. Z. Li, and M. PietikäInen. Facial expression recognition from near-infrared videos. *Image and Vision Computing*, 29(9):607–619, 2011.

[67] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. *Proc. Conf. Neural Information Processing Systems (NIPS)*, 2006.

[68] S. Gu, L. Zhang, W. Zuo, and X. Feng. Projective dictionary pair learning for pattern classification. In *Advances in neural information processing systems*, pages 793–801, 2014.

[69] U. Johansson, H. Linusson, T. Löfström, and H. Boström. Model-agnostic nonconformity functions for conformal classification. In *Proc. International Joint Conference on Neural Networks (IJCNN) 2017*, pages 2072–2079, 2017.

[70] G. wang and j. hwang and c. rose and f. wallace. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6, 2017.

[71] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[72] R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185, 2000.

[73] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[74] S. Matiz and K. E. Barner. Inductive conformal predictor for sparse coding classifiers: Applications to active learning for image classification. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), https://doi.org/10.1109/ICASSP.2019.8682740*, pages 3307–3311. IEEE, 2019.

[75] S. Matiz and K. E. Barner. Inductive conformal predictor for convolutional neural networks: Applications to active learning for image classification. *Pattern Recognition, https://doi.org/10.1016/j.patcog.2019.01.035*, 90:172–182, 2019.

[76] S. Matiz and K. E. Barner. Comformal prediction based active learning by linear regression optimization. *Neurocomputing (paper under revision)*, 2019.

[77] S. Matiz and K. E. Barner. Comformal prediction based active learning by nonlinear constrained optimization. *paper to be submitted (ongoing work)*, 2019.

[78] K. M. Ting, T. Washio, J. R. Wells, and F. T. Liu. Density estimation based on mass. In *2011 IEEE 11th International Conference on Data Mining*, pages 715–724, Dec 2011.

[79] Q. Wang, P. C. Yuen, and G. Feng. Semi-supervised metric learning via topology preserving multiple semi-supervised assumptions. *Pattern Recognition*, 46(9):2576–2587, 2013.

[80] Z. Zhang and M. M. Crawford. A batch-mode regularized multimetric active learning framework for classification of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6594–6609, 2017.

Appendix

COPYRIGHT NOTICE

This appendix contains the copyright notices for the contents of this dissertation.