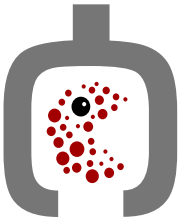


pacman

Probabilistic and Compositional Representations for Object Manipulation

FP7-IST-60918

1 March 2013 (36months)

DR1.1: Compositional Hierarchies of Object Categories Observed from Multiple Views

Mete Ozay, Vladislav Kramarev, Sebastian Zurek, U. Rusen Aktas, Maxime Adjigble, Mirela Popa, Carlos J. Rosales Gallegos, Aleš Leonardis, Jeremy Wyatt

School of Computer Science, University of Birmingham, United Kingdom.

⟨m.ozay@cs.bham.ac.uk⟩

Due date of deliverable:	2014-02-28
Actual submission date:	2014-02-28
Lead partner:	BHAM
Revision:	final
Dissemination level:	PU

This report describes the algorithms proposed regarding Deliverable D1.1.

Executive Summary

This report presents work carried out in WP1 on Compositional Hierarchies of object categories observed from multiple views. The work addresses Tasks 1.1 and 1.2, and supports Task 1.3. We first describe two approaches to learning 2D shape compositional hierarchies of object categories to incorporate visual information from multiple viewpoints as defined in Task 1.1. The work led to two publications; *i*) a conference publication which will be published in Proc. IEEE Conf. Robotics and Automation (ICRA), 2014 [5] (see Annex 2.1), and *ii*) a conference publication submitted to ECCV 2014 [1] (see Annex 2.2 for a Technical Report version of the conference paper). Regarding Task 1.2, a hierarchical compositional architecture is described which learns a vocabulary to capture 3D structural elements of the objects, where depth disparities constitute the first layer of the hierarchy. The work has been reported in a conference publication submitted to ICPR 2014 [3] (see Annex 2.3).

In addition, multi-modal information integration problem, which is addressed in Task 1.3, has been achieved by analyzing the theoretical principles of hierarchical consensus and collaborative learning algorithms. The work was initialized before the project started, and has been completed in the first year of the project. The work on hierarchical consensus learning has been reported in a conference publication submitted to ICPR 2014 [4] (see Annex 2.4). The proposed hierarchical collaborative learning algorithm and analyses have been introduced in a journal paper submitted to IEEE Transactions on Fuzzy Systems [6] given in Annex 2.5.

Role of Compositional Hierarchies of object categories observed from multiple views in PaCMan

In WP1, we focus on learning 2D compositional hierarchical models from multiple viewpoints (Task 1.1), and then on learning a 3D compositional shape vocabulary (Task 1.2). Results that are obtained from theoretical and experimental analyses of hierarchical consensus and collaborative learning algorithms can be used for the integration of multi-modal information in Task 1.3 and WP2. Incremental learning methods are considered in the proposed multiple view 2D Compositional Hierarchical architecture which will be used to process actively acquired data to support WP3. The proposed algorithms will be utilized in the grasping and dishwasher-scenario tasks which are described in WP4 and WP5.

Contribution to the PaCMan scenario

The proposed algorithms will be used to process visual information in the dishwasher-scenarios addressed in WP4 and WP5.

1 Tasks, objectives, results

1.1 Planned work

DR 1.1 is supposed to address Compositional hierarchies of object categories observed from multiple views. Planned work mainly concerns Task 1.1 regarding multi-view learning of compositional 2D models of object appearance, and Task 1.2 regarding Compositional 3D models of objects.

The objective of Task 1.1 is extending the current approach for learning 2D shape compositional hierarchies for multiple view object categorization and pose estimation by systematic incorporation of novel views. Additionally, a camera-robot setup was to be designed and implemented for acquisition of visual information from multiple viewpoints.

In Task 1.2, a 3D hierarchical compositional shape representation that captures statistically relevant structures of 3D objects was supposed to be designed by learning disparities (i.e. absolute depth with respect to the vergence point). In addition, a robot-head setup was supposed to be designed and implemented for acquiring 3D visual information with a verging system of stereo cameras.

Work on integration of multi-modal data, which supports Task 1.3, has already started in year 1, although only theoretical results are reported in deliverable DR 1.1.

1.2 Actual work performed

In this section, the main achievements related to the topic of this deliverable are briefly described. For detailed descriptions of the work performed the reader is referred to the papers attached in the annex to this deliverable.

1.2.1 Task 1.1 Multi-view learning of compositional 2D models of object appearance

Two hierarchical compositional architectures have been employed in order to learn 2D shape compositional hierarchies for multiple view object categorization and pose estimation.

In an IEEE ICRA paper given in Annex 2.1 [5], we propose a joint object pose estimation and categorization approach which extracts information about object poses and categories from the object parts and compositions constructed at different layers of a hierarchical object representation algorithm, namely Learned Hierarchy of Parts (LHOP) [2]. In the proposed approach, we first employ LHOP to learn hierarchical part libraries which represent entity parts and compositions across different object categories and views. Then, statistical and geometric features are extracted from the part realizations of the objects in the images in order to represent the information

about object pose and category at each different layer of the hierarchy. Unlike traditional approaches which consider specific layers of the hierarchies in order to extract information to perform specific tasks, we combine the information extracted at different layers to solve a joint object pose estimation and categorization problem using a generative-discriminative learning approach.

Descriptive models have been incorporated to compositional hierarchies using a graph theoretical approach introduced in a Technical Report and a conference paper which is submitted to ECCV, 2014 [1], and given in Annex 2.2. Two information theoretic algorithms are used for learning a vocabulary of compositional parts. In the proposed hybrid generative-descriptive learning model, statistical relationships between parts are quantified as the amount of information needed to describe a realization of a shape part given the realizations of other parts on 2D images. The statistical relationships are learned using a *Minimum Conditional Entropy Clustering* algorithm. Next, *contribution* of a part to representation of a shape in a part composition is *described* by measuring a *conditional description length* of the part given a compositional representation of the shape at a layer of the hierarchy. Then, part selection problem is defined as a Subgraph Isomorphism problem, and solved using an MDL principle. Finally, part compositions are constructed considering learned statistical relationships between parts and their conditional description length.

The proposed approach and algorithms are examined using a multiple view image dataset and two articulated image datasets. Experimental results show that CHOP can recognize and use part shareability property in the construction of vocabularies and inference trees. For instance, if parts of shapes encoded in a learned vocabulary and a new given shape, which will be used for incremental learning of vocabulary, are shareable, then the shareable parts can be used to improve the statistical relationships between learned parts, and minimize description length of parts and compositions in the CHOP. Additionally, junctions and closed curves observed at the shape boundaries can be detected as part realizations if they are shared among different articulated images.

A robot-head setup has also been designed for acquiring 3D visual information with a verging system of stereo cameras in BHAM. The stereo camera system consists of four firewire cameras. Additionally, there is one Primesense depth camera and an IR sensor. A turntable setup has been designed for acquisition of visual information from multiple viewpoints. The turntable setup consists of one rotary positioner, which is used for rotating a platform by 360^o, and a linear positioner, which is used to move the platform up-down by 225mm.

In order to acquire 3D visual scenes from different viewpoints, a robot system (Kuka arms and Schunk hands) and a number of Kinect sensors are used in UIBK. An experiment is designed for capturing images of objects

in an Ikea dataset from different viewpoints. One robot arm is equipped with a Kinect sensor, while the other arm is equipped with a wooden table. Views of an objects are obtained by moving an arm (with Kinect) at selected degrees around the object, while adjusting the position of the other arm in order to capture images at different scales.

Using the NUKLEI algorithm employed in Task 1.4, pose estimation is achieved by searching for the maximum of $p(w)$, where p accounts for the object's pose distribution, and w denotes a rigid transformation. Maximum-likelihood (ML) computations are performed using Monte Carlo methods. The ML pose $p(w)$ is computed via simulated annealing on a Markov chain. As $p(w)$ is likely to present a large number of narrow modes, we use a mixture of global and local proposals as a compromise between distributed exploration of the pose space and fine tuning of promising regions. The Markov chain is defined with a mixture of two local and global Metropolis-Hastings transition kernels. The location bandwidth of this kernel is set to a fraction of the size of the object, which in turn is computed as the standard deviation of input object points to their center of gravity. Its orientation bandwidth is set to a constant allowing for 5 degrees of deviation.

There are two experimental setups that use visual information at UNIPI related to the PaCMan project: *i*) a robotic platform, and *ii*) a sensorized grasp environment. The robotic platform is composed of an RGBD camera and two Kuka LWR attached to a rigid torso. The camera is mounted as the torso head and it is fixed. The camera-robot calibration uses the depth information to recognize a 3D part on the robot or in the environment with a known pose with respect to the torso. In a scenario, we perform haptic explorations with one of the arms acting as a probe over object surfaces. The object surface is acquired with the camera and modeled with a Gaussian process. The preliminary results successfully estimate the dynamic friction coefficient to be included in an adequate object representation. In another scenario, a sensorized grasp environment is composed of an RGBD camera, a led-based motion tracking system and a sensorized glove equipped with leds and intrinsic tactile sensors. In the scenario, we perform grasping experiments with a robotic hand attached to the forearm or directly with subject's human hand, in both cases using the sensorized globe. The camera-tracking system calibration is done similarly to the previous setup, using the depth information and a known object with leds. The prepared datasets consist of object and hand pose tracking data, contact points and the associated point cloud captured during grasping actions.

The Libhop C++ code developed by Prof. Leonardis' group at the University of Ljubljana has been a useful tool to investigate visual object categorization using compositional hierarchies. However, since the departure of the code's key software architect and developer, it has been difficult to maintain and extend the large C++ codebase in order to support the research work of the PaCMan project. Thus the code is being refactored to facilitate

further enhancements such as new algorithms, and to ease integration with other PaCMan software components. A documentation for users has been prepared, and a documentation for developers will also be produced.

1.2.2 Task 1.2 Compositional 3D models of objects

We have developed a framework for learning and recognition of a hierarchical compositional representation of 3D shapes in a conference paper submitted to ICPR 2014 [3] (see Annex 2.3). The elements of the first layer of the compositional hierarchy encode different disparities in range data. The framework subsequently learns layers of the hierarchy taking the most relevant compositions of parts from the previous layer. The complexity of the learned parts goes up with the number of layers. Parts start from features capturing relative depth to quite complex surface parts representing corners, and various convex and concave surface types.

1.2.3 Integration of Multi-modal Information

We have analyzed theoretical principles of hierarchical consensus and collaborative learning algorithms for integration of multi-modal information in a conference paper submitted to ICPR 2014 [4] in Annex 2.4, and a journal paper submitted to IEEE Transactions on Fuzzy Systems [6] in Annex 2.5. A stochastic distributed optimization algorithm is proposed to integrate multi-modal information obtained from different information channels of images. The proposed algorithm is used to achieve a consensus among different segmentation outputs obtained from segmentation algorithms by maximizing the joint probability of observing the segments at outputs of different segmentation algorithms. We will use the consensus algorithm for statistical feature binding in terms of joint probabilities between 2D and 3D parts. In addition, the correlations between shape parts of an object which are observed across multiple viewpoints of the object will be learned, and shape part deformations will be connected to viewpoint changes enabling predictive next-view planning using the proposed consensus algorithm.

We analyzed the relationship between shareability of features among different base-layer discriminative learning algorithms and the categorization performance in a hierarchical categorization algorithm given in Annex 2.5 [6]. Theoretical and experimental results show that the categorization performance increases as the feature shareability increases. In other words, if the parts are composed in order to increase the shareability (i.e. *a degree of collaboration*) among base-layer discriminative learning algorithms in a hierarchical architecture which employs a generative-discriminative approach, then the categorization performance of the hierarchy is greater than or equal to the best categorization performance provided by the base-layer algorithms. The results will be used for the design of discriminative parts

and compositions for multi-view learning of compositional 2D models, and for integration of 2D and 3D representations of objects to boost categorization performance.

1.3 Relation to the state-of-the-art

We examine the proposed generative-discriminative learning approach and the algorithms on two benchmark 2-D multi-view image datasets for joint object categorization and pose estimation in an ICRA paper [5] (see Annex 2.1). The proposed approach and the algorithms outperform state-of-the-art classification, regression and feature extraction algorithms, such as Support Vector Machines, Support Vector Regression, Lasso, Logistic Regression and Histogram of Oriented Gradients. In addition, the experimental results shed light on the relationship between statistical and geometric properties of the part realizations observed at different layers of the hierarchy, and object categorization vs. pose estimation performance.

Compositional Hierarchy of Parts proposed in the ECCV 2014 paper [1] (see Annex 2.2) is the first system to fully encode and infer compositional parts of objects using hybrid generative-descriptive learning models within a graph-based hierarchical compositional framework to the best of our knowledge.

In a conference paper submitted to ICPR 2014 [3] (see Annex 2.2), we have tested our 3D hierarchical compositional representation for a 3D object categorization problem. We achieved categorization performance that is close to state-of-the-art performance on standard object categorization datasets using four layers of features.

The proposed hierarchical consensus learning approach is used for Semi-supervised Segmentation Fusion of multi-spectral images in a paper submitted to ICPR 2014 [4](see Annex 2.3). The experimental results show that the proposed algorithms perform better than the individual state-of-the art clustering and image segmentation algorithms, such as k-means, Mean Shift and Graph Cut Segmentation. The hierarchical categorization algorithm proposed in [6] (see Annex 2.4) bridges the gap between finite and large sample categorization error of the nearest neighbor algorithm, which is the best achievable categorization error by any categorization algorithm for a large number of training samples. Experiments on the image categorization datasets show that the proposed algorithm performs better than the state of the art hierarchical learning algorithms such as, Adaboost, Random Subspace and Rotation Forest, which are given in Annex 2.5 [6].

References

[1] U. R. Aktas, M. Ozay, A. Leonardis, and J. Wyatt, “A graph theoretic approach for object shape representation in compositional hierarchy of

parts using an hybrid generative-descriptive model," in *Technical Report, an extended version is submitted to The European Conference on Computer Vision (ECCV)*, 2014.

[2] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[3] V. Kramarev, S. Zurek, J. L. Wyatt, and A. Leonardis, "Object categorization from range images using a hierarchical compositional representation," in *submitted to 22nd International Conference on Pattern Recognition (ICPR)*, 2014.

[4] M. Ozay, "Semi-supervised segmentation fusion of multi-spectral and aerial images," in *submitted to 22nd International Conference on Pattern Recognition (ICPR)*, 2014.

[5] M. Ozay, K. Walas, and A. Leonardis, "A hierarchical approach for joint multi-view object pose estimation and categorization," in *Proc. IEEE Conf. Robotics and Automation*, 2014.

[6] M. Ozay and F. T. Yarman Vural, "A new fuzzy stacked generalization technique and analysis of its performance," *submitted to IEEE Transactions on Fuzzy Systems*, 2014.

2 Annexes

2.1 A Hierarchical Approach for Joint Multi-view Object Pose Estimation and Categorization

Bibliography M. Ozay, K. Walas, and A. Leonardis A Hierarchical Approach for Joint Multi-view Object Pose Estimation and Categorization in Proc. IEEE Conf. Robotics and Automation, 2014.

Abstract We propose a joint object pose estimation and categorization approach which extracts information about object poses and categories from the object parts and compositions constructed at different layers of a hierarchical object representation algorithm, namely Learned Hierarchy of Parts (LHOP). In the proposed approach, we first employ the LHOP to learn hierarchical part libraries which represent entity parts and compositions across different object categories and views. Then, we extract statistical and geometric features from the part realizations of the objects in the images in order to represent the information about object pose and category at each different layer of the hierarchy. Unlike the traditional approaches which consider specific layers of the hierarchies in order to extract information to perform specific tasks, we combine the information extracted at different layers to solve a joint object pose estimation and categorization problem using distributed optimization algorithms. We examine the proposed generative-discriminative learning approach and the algorithms on two benchmark 2-D multi-view image datasets. The proposed approach and the algorithms outperform state-of-the-art classification, regression and feature extraction algorithms. In addition, the experimental results shed light on the relationship between object categorization, pose estimation and the part realizations observed at different layers of the hierarchy.

Relation to WP The paper addresses Compositional Hierarchies of object categories observed from multiple views (Task 1.1).

2.2 A Graph Theoretic Approach for Object Shape Representation in Compositional Hierarchies using a Hybrid Generative-Descriptive Model

Bibliography Umit Rusen Aktas, Mete Ozay, Aleš Leonardis and Jeremy Wyatt, A Graph Theoretic Approach for Object Shape Representation in Compositional Hierarchies using a Hybrid Generative-Descriptive Model Technical report, submitted to European Conference on Computer Vision, 2014.

Abstract A graph theoretical approach is proposed for object shape representation in a hierarchical compositional architecture called Compositional Hierarchy of Parts (CHOP). In the proposed approach, vocabulary learning is performed using a hybrid generative-descriptive model. Two information theoretic algorithms are used for learning a vocabulary of compositional parts. First, statistical relationships between parts are quantified as the amount of information needed to describe a realization of a shape part given the realizations of other parts on 2D images. The statistical relationships are learned using a *Minimum Conditional Entropy Clustering* algorithm. Second *contribution* of a part to representation of a shape in a part composition is *described* by measuring a *conditional description length* of the part given a compositional representation of the shape at a layer of the hierarchy. Then, part selection problem is defined as a Subgraph Isomorphism problem, and solved using an MDL principle. Finally, part compositions are constructed considering learned statistical relationships between parts and their conditional description length.

The proposed approach and algorithms are examined using a multiple view image dataset and two articulated image datasets. Experimental results show that CHOP can recognize and use part shareability property in the construction of vocabularies and inference trees. For instance, if parts of shapes encoded in a learned vocabulary and a new given shape, which will be used for incremental learning of vocabulary, are shareable, then the shareable parts can be used to improve the statistical relationships between learned parts, and minimize description length of parts and compositions in the CHOP. Additionally, junctions and closed curves observed at the shape boundaries can be detected as part realizations if they are shared among different articulated images.

Relation to WP The paper addresses a graph theoretical approach for representation of object shapes in a hierarchical compositional architecture using multiple view and articulated 2D images (Task 1.1).

2.3 Object Categorization from Range Images using a Hierarchical Compositional Representation

Bibliography V. Kramarev, S. Zurek, J. Wyatt, and A. Leonardis Object Categorization from Range Images using a Hierarchical Compositional Representation submitted to ICPR 2014.

Abstract This paper proposes a novel hierarchical compositional representation of 3D shape that can accommodate a large number of object categories and enables efficient learning and inference. The hierarchy starts with simple pre-defined parts on the first layer, after which subsequent layers are learned recursively by taking the most statistically significant compositions of parts from the previous layer. Our representation is able to scale because of its very economical use of memory and because subparts of the representation are shared. We apply our representation to 3D multi-class object categorization. Object categories are represented by histograms of compositional parts, which are then used as inputs to an SVM classifier. We present results for two datasets, Aim@Shape and the Washington RGB-D Object Dataset, and demonstrate the competitive performance of our method.

Relation to WP The paper addresses Task 1.3 and presents an algorithm for learning of subsequent layers of the hierarchical 3D shape vocabulary.

2.4 Semi-supervised Segmentation Fusion of Multi-spectral and Aerial Images

Bibliography M. Ozay, Semi-supervised Segmentation Fusion of Multi-spectral and Aerial Images, submitted to ICPR 2014.

Abstract A Semi-supervised Segmentation Fusion algorithm is proposed using consensus and distributed learning. The aim of Unsupervised Segmentation Fusion (USF) is to achieve a consensus among different segmentation outputs obtained from different segmentation algorithms by computing an approximate solution to the NP problem with less computational complexity. Semi-supervision is incorporated in USF using a new algorithm called Semi-supervised Segmentation Fusion (SSSF). In SSSF, side information about the co-occurrence of pixels in the same or different segments is formulated as the constraints of a convex optimization problem. The results of the experiments employed on artificial and real-world benchmark multi-spectral and aerial images show that the proposed algorithms perform better than the individual state-of-the art segmentation algorithms.

Relation to WP The paper considers multi-modal information integration problem which is addressed in Task 1.3. The proposed consensus learning algorithms will be used for the integration of 2D and 3D information obtained from compositional hierarchies.

2.5 A New Fuzzy Stacked Generalization Technique and Analysis of its Performance

Bibliography M. Ozay, F. T. Yarman Vural, A New Fuzzy Stacked Generalization Technique and Analysis of its Performance, submitted to IEEE Transactions on Fuzzy Systems, 2014.

Abstract A new Stacked Generalization method which employs a hierarchical distance learning strategy in a two-layer ensemble learning architecture called Fuzzy Stacked Generalization (FSG) is proposed. At the base-layer of FSG, fuzzy k-Nearest Neighbor (k-NN) classifiers map their own input feature vectors into the posteriori probabilities. At the meta-layer, a fuzzy k-NN classifier learns a distance function by minimizing the difference between the large sample and N-sample classification error using the estimated posteriori probabilities. In the FSG, the feature space of each base-layer classifier is designed to gain an expertise on a specific property of the dataset, whereas the meta-layer classifier learns the degree of accuracy of the decisions of the base-layer classifiers. Experimental results obtained using the artificial datasets show that the classification performance of the FSG depends on diversity and cooperation of the classifiers rather than the classification performances of the individual base-layer classifiers. A weak base-layer classifier may boost the overall performance of the FSG more than a strong classifier, if it is capable of recognizing the samples, which are not recognized by the rest of the classifiers. The cooperation among the base-layer classifiers is quantified by introducing a shearability measure. The effect of the shearability on the performance is investigated on the artificial datasets. Experiments on the real datasets show that FSG performs better than the state of the art ensemble learning algorithms such as, Adaboost, Random Subspace and Rotation Forest.

Relation to WP The paper considers multi-modal information integration problem which is addressed in Task 1.3. The proposed collaborative learning algorithm will be used for the integration of 2D and 3D information obtained from compositional hierarchies. In addition, the results obtained from the analyses on feature shareability will be used for the development of disriminative learning models in 2D hierarchical compositional architectures for categorization and pose estimation.

A Hierarchical Approach for Joint Multi-view Object Pose Estimation and Categorization

Mete Ozay[1], Krzysztof Walas[1,2] and Aleš Leonardis[1]

***Abstract*— We propose a joint object pose estimation and categorization approach which extracts information about object poses and categories from the object parts and compositions constructed at different layers of a hierarchical object representation algorithm, namely Learned Hierarchy of Parts (LHOP) [7]. In the proposed approach, we first employ the LHOP to learn hierarchical part libraries which represent entity parts and compositions across different object categories and views. Then, we extract statistical and geometric features from the part realizations of the objects in the images in order to represent the information about object pose and category at each different layer of the hierarchy. Unlike the traditional approaches which consider specific layers of the hierarchies in order to extract information to perform specific tasks, we combine the information extracted at different layers to solve a joint object pose estimation and categorization problem using distributed optimization algorithms. We examine the proposed generative-discriminative learning approach and the algorithms on two benchmark 2-D multi-view image datasets. The proposed approach and the algorithms outperform state-of-the-art classification, regression and feature extraction algorithms. In addition, the experimental results shed light on the relationship between object categorization, pose estimation and the part realizations observed at different layers of the hierarchy.**

I. Introduction

The field of service robots aims to provide robots with functionalities which allow them to work in man-made environments. For instance, the robots should be able to categorize objects and estimate the pose of the objects to accomplish various robotics tasks, such as grasping objects [14]. Representation of object categories enables the robot to further refine the grasping strategy by giving context to the search for the pose of the object [15].

In this paper, we propose a joint object categorization and pose estimation approach which extract information about statistical and geometric properties of object poses and categories extracted from the object parts and compositions that are constructed at different layers of the Learned Hierarchy of Parts (LHOP) [7], [8], [9].

In the proposed approach, we first employ LHOP [7], [8] to learn hierarchical part libraries which represent object parts and compositions across different object categories and views as shown in Fig. 1. Then, we extract statistical and geometric features from the part realizations of the objects in the images in order to represent the information about the object pose and category at each different layer of the hierarchy. We propose two novel feature extraction algorithms, namely Histogram of Oriented Parts (HOP) and Entropy of Part Graphs. HOP features measure local distributions of global orientations of part realizations of objects at different layers of a hierarchy. On the other hand, Entropy of Part Graphs provides information about the statistical and geometric structure of object representations by measuring the entropy of the relative orientations of parts. In addition, we compute a Histogram of Oriented Gradients (HOG) [5] of part realizations in order to obtain information about the co-occurrence of the gradients of part orientations.

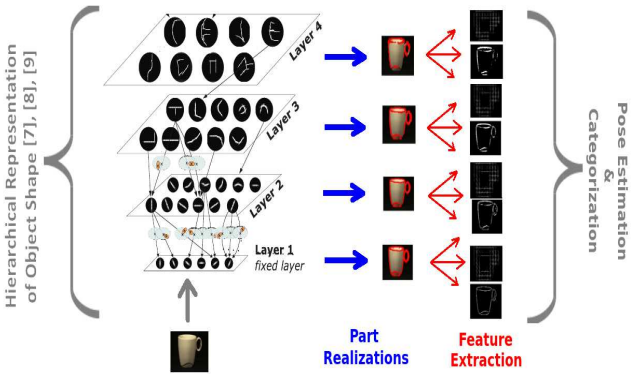


Fig. 1: Combination of features extracted from part realizations detected at different layers of LHOP.

Unlike traditional approaches which extract information from the object representations at specific layers of the hierarchy to accomplish specific tasks, we combine the information extracted at different layers to solve a joint object pose estimation and categorization problem using a distributed optimization algorithm. For this purpose, we first formulate the joint object pose estimation and categorization problem as a sparse optimization problem called Group Lasso [19]. We consider the pose estimation problem as a sparse regression problem and the object categorization problem as a multi-class logistic regression problem using Group Lasso. Then, we solve the optimization problems using a distributed and parallel optimization algorithm called the Alternating Direction Method of Multipliers (ADMM) [1].

In this work, we extract information on object poses and categories from 2-D images to handle the cases where 3-

This work was supported in part by the European Commission project PaCMan EU FP7-ICT, 600918.

[1]Mete Ozay, Krzysztof Walas and Aleš Leonardis are with School of Computer Science, University of Birmingham, Edgbaston B15 2TT Birmingham, United Kingdom {m.ozay,walask,a.Leonardis}@cs.bham.ac.uk

[2]Krzysztof Walas is also with Department of Electrical Engineering, Poznan University of Technology, ul. Piotrowo 3a, 60-965 Poznan, Poland krzysztof.walas@put.poznan.pl

D sensing may not be available or may be unreliable (e.g. glass, metal objects). We examine the proposed approach and the algorithms on two benchmark 2-D multiple-view image datasets. The proposed approach and the algorithms outperform state-of-the-art Support Vector Machine and Regression algorithms. In addition, the experimental results shed light on the relationship between object categorization, pose estimation and the part realizations observed at different layers of the hierarchy.

In the next section, related work is reviewed and the novelty of our proposed approach is summarized. In Section II, a brief presentation of the hierarchical compositional representation is given. Feature extraction algorithms are introduced in Section III. The joint object pose estimation and categorization problem is defined, and two algorithms are proposed to solve the optimization problem in Section IV. Experimental analyses are given in Section V. Section VI concludes the paper.

A. Related Work and Contribution

In the field of computer vision the problem of object categorization and pose estimation is studied thoroughly and some of the approaches are proliferating to the robotics community. With an advent of devices based on PrimeSense sensors, uni-modal 3-D or multi-modal integration of 2-D and 3-D data (e.g. rgb-d data) have been widely used by robotics researchers [13]. However, 3-D sensing may not be available or reliable due to limitations of object structures, lighting resources and imaging conditions in many cases where single or multiple view 2-D images are used for categorization and pose estimation [3], [4], [20]. In [20], a probabilistic approach is proposed to estimate the pose of a known object using a single image. Collet et al. [3] build 3D models of objects using SIFT features extracted from 2D images for robotic manipulation, and combine single image and multiple image object recognition and pose estimation algorithms in a framework in [4].

A promising approach to the object categorization and the scene description is the use of hierarchical compositional architectures [7], [9], [15]. Compositional hierarchical models are constructed for object categorization and detection using single images in [7], [9]. Multiple view images are used for pose estimation and categorization using a hierarchical architecture in [15]. In the aforementioned approaches, the tasks are performed using either discriminative or generative top-down or bottom-up learning approaches in architectures. For instance, Lai et al. employ a top-down categorization and pose estimation approach in [15], where a different task is performed at each different layer of the hierarchy. Note that, a categorization error occurring at the top-layer of the hierarchy may propagate to the lower layer and affect the performance of other tasks such as pose estimation in this approach. In our proposed approach, we first construct generative representations of object shapes using LHOP [7], [8], [9]. Then, we train discriminative models by extracting features from the object representations. In addition, we propose a new method, which enables us to combine the information extracted at each different layer of the hierarchy, for joint categorization and pose estimation of objects. We avoid the propagation of errors of performing multiple tasks through the layers and enable the shareability of parts among layers by the employment of optimization algorithms in each layer in a parallel and distributed learning framework.

The novelty of the proposed approach and the paper can be summarized as follows;

1) In this work, the Learned Hierarchy of Parts (LHOP) is employed in order to learn a hierarchy of parts using the shareability of parts across different views as well as different categories [7], [8].
2) Two novel feature extraction algorithms, namely Histogram of Oriented Parts (HOP) and Entropy of Part Graphs, are proposed in order to obtain information about the statistical and geometric structure of objects' shapes represented at different layers of the hierarchy using part realizations.
3) The proposed generative-discriminative approach enables us to combine the information extracted at different layers in order to solve a joint object pose estimation and categorization problem using a distributed and parallel optimization algorithm. Therefore, this approach also enables us to share the parts among different layers and avoid the propagation of object categorization and pose estimation errors through the layers.

II. Learned Hierarchy of Parts

In this section, Learned Hierarchy of Parts (LHOP)[7], [8] is briefly described. In LHOP, the object recognition process is performed in a hierarchy starting from a feature layer through more complex and abstract interpretations of object shapes to an object layer. A learned vocabulary is a recursive compositional representation of shape parts. Unsupervised bottom-up statistical learning is encompassed in order to obtain such a description.

Shape representations are built upon a set of compositional parts which at the lowest layer use atomic features, e.g. Gabor features, extracted from image data. The object node is a composition of several child nodes located at one layer lower in the hierarchy, and the composition rule is recursively applied to each of its child nodes to the lowest layer Γ_1. All layers together form a hierarchically encoded vocabulary $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \ldots \cup \Gamma_L$. The entire vocabulary Γ is learned from the training set of images together with the vocabulary parameters [8].

The parts in the hierarchy are defined recursively in the following way. Each part in the l^{th} layer represents the spatial relations between its constituent subparts from the layer below. Each composite part $\mathcal{P}_k^l$ constructed at the l^{th} layer is characterized by a central subpart $\mathcal{P}_{central}^{l-1}$ and a list of remaining subparts with their positions relative to the center as

$$\mathcal{P}_k^l = (\mathcal{P}_{central}^{l-1}, \{(\mathcal{P}_j^{l-1}, \boldsymbol{\mu}_j, \Sigma_j)\}j), \quad (1)$$

where $\boldsymbol{\mu}_j = (x_j, y_j)$ denotes the relative position of the subpart $\mathcal{P}_j^{l-1}$, while Σ_j denotes the allowed variance of its position around (x_j, y_j).

III. Feature Extraction from Learned Parts

LHOP provides information about different properties of objects, such as poses, orientations and category memberships, at different layers [7]. For instance, the information on shape parts, which are represented by edge structures and textural patterns observed in images, is obtained using Gabor features at the first layer L_1. In the second and the following layers, compositions of parts are constructed according to the co-occurrence of part realizations that are detected in the images among different views of the objects and across different object categories. In other words, a library of object parts and compositions is learned jointly for all object views and categories.

In order to obtain information about statistical and geometric properties of parts, we extract three types of features from the part realizations detected at each different layer of the LHOP.

A. Histogram of Orientations of Parts

Histograms of orientations of parts are computed in order to extract information on the co-occurrence of orientations of the parts across different poses of objects. Part orientations are computed according to a coordinate system of an image I whose origin is located at the center of the image I, and the axes of the coordinate system are shown with blue lines in Figure 2.

If we define $p_k^l, \forall k = 1, 2, \ldots, K, \forall l = 1, , 2 \ldots, L$ as the realization of the k^{th} detected part in the l^{th} layer at an image coordinate (x_k, y_k) of I, then its orientation with respect to the origin of the coordinate system is computed as

$$\theta_{k,l} = \arctan(\frac{y_k}{x_k}).$$

Then, the image I is partitioned into M cells $\{I_m\}_{m=1}^{M}$, and histograms of the part orientations $\{\theta_{k,l}\}_{k=1}^{K'}$ of the part realizations $\{p_{k,l}\}_{k=1}^{K'}$ that are located in each cell I_m are computed. The aggregated histogram values are considered as variables of a D_p dimensional feature vector $\mathbf{f}_{hop}^l \in \mathbb{R}^{D_p}$.

B. Histogram of Oriented Gradients of Parts

In addition to the computation of histograms of orientations of part realizations $p_k^l, \forall k = 1, 2, \ldots, K, \forall l = 1, 2, \ldots, L$, we compute histogram of oriented gradients (HOG) [5] of p_k^l in order to extract information about the distribution of gradient orientations of $p_k^l, \forall k, l$. We denote the HOG feature vector extracted using $\{p_k^l\}_{k=1}^{K}$ in the l^{th} layer as $\mathbf{f}_{hog}^l \in \mathbb{R}^{D_h}$, where D_h is the dimension of the HOG feature vector. The details of the implementation of HOG feature vectors are given in Section V.

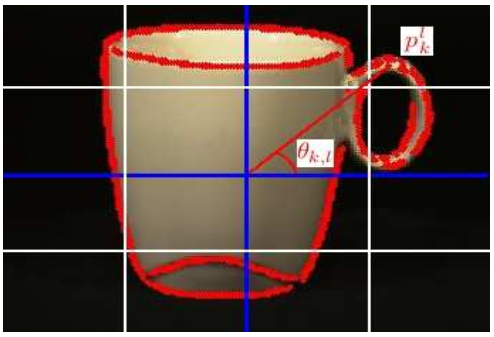


Fig. 2: An image is partitioned into cells for the computation of histograms of orientations of parts. A part realization p_k^l is depicted with a red point and associated to a part orientation degree $\theta_{k,l}$.

C. The Entropy of Part Graphs

We measure the statistical and structural properties of relative orientations of part realizations by measuring the complexity of a graph of parts. Mathematically speaking, we define a weighted undirected graph $G_l := (E_l, V_l)$ in the l^{th} layer, where $V_l := \{p_k^l\}$ is the set of part realizations, $E_l := \{e_{k',k}\}_{k',k=1}^{K}$ is the set of edges, where each edge $e_{k',k}$ that connects the part realizations $p_{k'}^l$ and p_k^l is associated to an edge weight $w_{k',k}$, which is defined as

$$w_{k',k} := \arccos(\frac{\mathbf{pos}_{k'} \cdot \mathbf{pos}_k}{\|\mathbf{pos}_{k'}\|_2 \|\mathbf{pos}_k\|_2}),$$

where $\mathbf{pos}_k := (x_k, y_k)$ is the position vector of $p_{k'}^l$, $\|\cdot\|_2$ is the ℓ_2 norm or Euclidean norm, and $\mathbf{pos}_{k'} \cdot \mathbf{pos}_k$ is the inner product of $\mathbf{pos}_{k'}$ and $\mathbf{pos}_k$. In other words, the edge weights are computed according to the orientations of parts relative to each other.

We measure the complexity of the weighted graph by computing its graph entropy. First, we compute the normalized weighted graph Laplacian $\mathcal{L}$ [6], [16] as

$$\mathcal{L} = \frac{1}{K(K-1)}(\mathcal{D} - \mathcal{W}),$$

where $\mathcal{W} \in \mathbb{R}^{K \times K}$ is a weighted adjacency matrix or a matrix of weights $w_{k',k}$, and $\mathcal{D} \in \mathbb{R}^{K \times K}$ is a diagonal matrix with members $\mathcal{D}_{k,k} := \sum_{k'=1}^{K} w_{k',k}$. Then, we compute the von Neumann entropy of G_l [6], [16] as

$$S(G_l) = -\operatorname{Tr}(\mathcal{L} \log_2 \mathcal{L}) \quad (2)$$

$$= -\sum_{k=1}^{K} \nu_k, \quad (3)$$

where $\nu_1 \geq \nu_2 \geq \ldots \geq \nu_k \geq \ldots \geq \nu_K = 0$ are the eigenvalues of $\mathcal{L}$, $\operatorname{Tr}(\mathcal{L} \log_2 \mathcal{L})$ is the trace of the matrix product $\mathcal{L} \log_2 \mathcal{L}$ and $0 \log_2 0 = 0$. We use $S(G_l)$ as a feature variable $f_{ent}^l := S(G_l)$.

IV. Combination of Information Obtained at Different Layers of LHOP for Joint Object Pose Estimation and Categorization

In hierarchical compositional architectures, a different object property, such as object shape, pose and category, is represented at a different layer of a hierarchy in a vocabulary [15]. According the structures of the abstract representations of the properties, i.e. vocabularies, recognition processes have been performed using either a bottom-up [7], [8] or top-down [15] approach. It's worth noting that the information in the representations are distributed among the layers in the vocabularies. In other words, the information about the category of an object may reside at the lower layers of the hierarchy instead of the top layer. In addition, lower layer atomic features, e.g. oriented Gabor features, provide information about part orientations which can be used for the estimation of pose and view-points of objects at the higher layers. Moreover, the relationship between the pose and category of an object is bi-directional. Therefore, an information integration approach should be considered in order to avoid the propagation of errors that occur in multi-task learning and recognition problems such as joint object categorization and pose estimation, especially when only one of the bottom-up and top-down approaches is implemented.

For this purpose, we propose a generative-discriminative learning approach in order to combine the information obtained at each different layer of LHOP using the features extracted from part realizations. We represent the features defining a $D_p + D_h + 1$ dimensional feature vector $\mathbf{f}^l = (\mathbf{f}^l_{hop}, \mathbf{f}^l_{hog}, f^l_{ent})$. The feature vector $\mathbf{f}^l$ is computed for each training and test image, therefore we denote the feature vector of the i^{th} image I_i as $\mathbf{f}^l_i$, $\forall i = 1, 2, \ldots, N$, in the rest of the paper.

We combine the feature vectors extracted at each l^{th} layer for object pose estimation and categorization under the following Group Lasso optimization problem [19]

$$\text{minimize} \quad \|\mathcal{F}\boldsymbol{\omega} - \mathbf{z}\|_2^2 + \lambda \sum_{l=1}^{L} \|\boldsymbol{\omega}_l\|_2, \tag{4}$$

where $\|\cdot\|_2^2$ is the squared ℓ_2 norm, $\lambda \in \mathbb{R}$ is a regularization parameter, $\boldsymbol{\omega}_l$ is the weight vector computed at the l^{th} layer, $\mathcal{F} \in \mathbb{R}^{N \times L}$ is a matrix of feature vectors $\mathbf{f}^l_i$, $\forall i = 1, 2, \ldots, N$, $\forall l = 1, 2, \ldots, L$ and $\mathbf{z} = (z_1, z_2, \ldots, z_N)$ is a vector of target variables $z_i \in \mathbb{R}$, $\forall i = 1, 2, \ldots, N$. More specifically, $z_i \in \Omega$ where Ω is a set of object poses, i.e. object orientation degrees, in a pose estimation problem.

We solve (4) using a distributed optimization algorithm called Alternating Direction Method of Multipliers [1]. For this purpose, we first re-write (4) in the ADMM form as follows

$$\begin{aligned} &\text{minimize} && \|\mathcal{F}\boldsymbol{\phi} - \mathbf{z}\|_2^2 + \lambda \sum_{l=1}^{L} \|\boldsymbol{\omega}_l\|_2 \\ &\text{subject to} && \boldsymbol{\omega}_l - \hat{\boldsymbol{\phi}}_l = \mathbf{0}\ , l = 1, 2, \ldots, L\ , \end{aligned} \tag{5}$$

where $\hat{\boldsymbol{\phi}}_l$ is the local estimate of the global variable $\boldsymbol{\phi}$ for $\boldsymbol{\omega}_l$ at the l^{th} layer. Then, we solve (5) in the following three steps [1], [18],

1) At each layer l, we compute $\boldsymbol{\omega}_l^{t+1}$ as

$$\boldsymbol{\omega}_l^{t+1} := \underset{\boldsymbol{\omega}_l}{\operatorname{argmin}} \Big(\rho \|\boldsymbol{\mu}_l^t\|_2^2 + \lambda \|\boldsymbol{\omega}_l\|_2 \Big), \tag{6}$$

where $\boldsymbol{\mu}_l^t = \mathcal{F}_l(\boldsymbol{\omega}_l - \boldsymbol{\omega}_l^t) - \bar{\boldsymbol{\phi}}^t + \mathbf{a}^t + \overline{\mathcal{F}_l\boldsymbol{\omega}_l}^t$, $\rho > 0$ is a penalty parameter, $\overline{\mathcal{F}_l\boldsymbol{\omega}_l}^t = \frac{1}{L}\sum_{l=1}^{L} \mathcal{F}_l\boldsymbol{\omega}_l^t$, $\bar{\boldsymbol{\phi}}^t$ is the average of $\boldsymbol{\phi}_l^t$, $\forall l = 1, \ldots, L$, and $\mathbf{a}^t$ is a vector of scaled dual optimization variables computed at an iteration t.

2) Then we update $\hat{\boldsymbol{\phi}}_l$ as

$$\hat{\boldsymbol{\phi}}_l^{t+1} := \frac{1}{L + \rho} \Big(\mathbf{z} + \rho \overline{\mathcal{F}_l\boldsymbol{\omega}_l}^{t+1} + \rho \mathbf{a}^t \Big). \tag{7}$$

3) Finally, $\mathbf{a}$ is updated as

$$\mathbf{a}^{t+1} := \mathbf{a}^t + \overline{\mathcal{F}_l\boldsymbol{\omega}_l}^t - \hat{\boldsymbol{\phi}}_l^{t+1}. \tag{8}$$

These three steps are iterated until a halting criterion, such as $t \geq T$ for a given termination time T, is achieved. Implementation details are given in the next section.

In a C class object categorization problem, $z_i \in \{1, 2, \ldots, c, \ldots, C\}$ is a category variable. In order to solve this problem, we employ *1-of-C* coding for sparse logistic regression as

$$P(z_i^c = 1 | \mathbf{f}_i) \quad = \quad \frac{\exp(h_j(\mathbf{f}_i))}{1 + \exp(h_c(\mathbf{f}_i))}, \tag{9}$$

where $h_c(\mathbf{f}_i) = \mathbf{f}_i \cdot \boldsymbol{\omega}^{\boldsymbol{c}}$, $\boldsymbol{\omega}^{\boldsymbol{c}}$ is a weight vector associated to the c^{th} category, $z_i^c = 1$ if $z_i = c$, $\forall i = 1, 2, \ldots, N$. Then, we define the following optimization problem

$$\text{minimize} \quad -\sum_{l=1}^{L}\sum_{i=1}^{N} loss_l(i) + \lambda \|\boldsymbol{\omega}^{\boldsymbol{c}}\|_1, \tag{10}$$

where $loss_l(i) = z_i^c h_c(\mathbf{f}_i) - \log\Big(\exp(h_c(\mathbf{f}_i)) + 1\Big)$. In order to solve (10), we employ the three update steps given above with two modifications. First, we solve (6) for the ℓ_1 norm in the last regularization term $\lambda\|\boldsymbol{\omega}_l\|_1$ instead of the ℓ_2 norm. Second, we employ the logistic regression loss function in the computation of $\hat{\boldsymbol{\phi}}_l$ as

$$\hat{\boldsymbol{\phi}}_l^{t+1} := \underset{\boldsymbol{\phi}_l}{\operatorname{argmin}} \Big(\rho \|\boldsymbol{\phi}_{\boldsymbol{l}} - \overline{\mathcal{F}_l\boldsymbol{\omega}_l}^{t+1} - \mathbf{a}^t\|_2 + \log(1 + \exp -(L\boldsymbol{\phi}_l)) \Big). \tag{11}$$

In the training phase of the pose estimation algorithm, we compute the solution vector $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_L\}$ using training data. In the test phase, we employ the solution vector $\boldsymbol{\omega}$ on a given test feature vector $\mathbf{f}_i$ of the part realizations of an object to estimate its pose as

$$\hat{z}_i = \mathbf{f}_i \cdot \boldsymbol{\omega}.$$

In the categorization problem, we predict the category label $\hat{z}_i$ of an object in the i^{th} image as

$$\hat{z}_i = \underset{c}{\operatorname{argmax}}\ \hat{z}_i^c.$$

V. EXPERIMENTS

We examine our proposed approach and algorithms on two benchmark object categorization and pose estimation datasets, which are namely the Amsterdam Library of Object Images (ALOI) [10] and the Columbia Object Image Library (COIL-100) [17]. We have chosen these two benchmark datasets for two main reasons. First, images of objects are captured by rotating the objects on a turntable by regular orientation degrees which enable us to analyze our proposed algorithm for multi-view object pose estimation and categorization in uncluttered scenes. Second, object poses and categories are labeled within *acceptable* precision which is important to satisfy the statistical stability of training and test samples and their target values. In our experiments, we also re-calibrated labels of pose and rotation values of the objects that are mis-recorded in the datasets.

We select the bin size ($bSize$) of the histograms and cell size M of HOP (see Section III-A) and HOG features (see Section III-B) by greedy search on the parameter set $\{8, 16, 32, 64\}$, and take the *optimal* $\hat{bSize}$ and $\hat{M}$ which minimizes pose estimation and categorization errors in pose estimation and categorization problems using training datasets, respectively. In the employment of optimization algorithms, we compute $\lambda = \alpha\lambda_{\max}$, where $\lambda_{\max} = \|\mathcal{F}\boldsymbol{\omega}\|_\infty$, $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_L)$, $\|\cdot\|_\infty$ is ℓ_∞ norm and α parameter is selected from the set $\{10^{-6}, 10^{-5}, \ldots, 10^{1}\}$ using greedy search by minimizing training error of object pose estimation and categorization as suggested in [1]. In the implementation of LHOP, we learn the compositional hierarchy of parts and compute the part realizations for $L = 1, 2, 3, 4$ [7].

In the experiments, pose estimation and categorization performances of the proposed algorithms are compared with state-of-the-art Support Vector Regression (SVR), Support Vector Machines (SVM) [2], Lasso and Logistic regression algorithms [12] which use the state-of-the-art HOG features [5] extracted from the images as considered in [11]. In the results, we refer to an implementation of SVM with HOG features as SVM-HOG, SVM with the proposed LHOP features as SVM-LHOP, SVR with HOG features as SVR-HOG, SVR with the proposed LHOP features as SVR-LHOP, Lasso with HOG features as L-HOG, Logistic Regression with HOG features as LR-HOG, Lasso with LHOP features as L-LHOP, Logistic Regression with LHOP features as LR-LHOP.

We use RBF kernels in SVR and SVM. The kernel width parameter σ is searched in the interval $\log(\sigma) \in [-10, 5]$ and the SVR cost penalization parameter ϵ is searched in the interval $\log(\epsilon) \in [-10, 5]$ using the training datasets.

A. Experiments on Object Pose Estimation

We have conducted two types of experiments for object pose estimation, namely *Object-wise* and *Category-wise* Pose Estimation. We analyze the sharability of the parts across different views of an object in Object-wise Pose Estimation experiments. In Category-wise Pose Estimation experiments, we analyze incorporation of category information to sharability of parts in the LHOP and to pose estimation performance.

1) Experiments on Object-wise Pose Estimation: In the first set of experiments, we consider the objects belonging to each different category, individually. For instance, we select $\aleph^o_{tr} = 4$ objects for training and $\aleph^o_{te} = 1$ objects for testing using objects belonging to **cups** category. The ID numbers of the objects and their category names are given in Table I. For each object, we have 72 object instances each of which represents an orientation of the object $z_i = \Theta_i$ on a turntable rotated with $\Theta_i \in \Omega$ and $\Omega = \{0°, 5°, 10°, \ldots, 355°\}$.

In the experiments, we first analyze the variation of part realizations and feature vectors across different orientations of an object. We visualize the features $\mathbf{f}^l_{hop}$, $\mathbf{f}^l_{hog}$ and f^l_{ent} in Figure 3 for a cup which is oriented with $\Theta \in \{20°, 60°, 120°, 180°, 240°, 280°, 340°\}$ and for each $l = 1, 2, 3, 4$. In the first row at the top of the figure, the change of f^l_{ent} is visualized $\forall l$. In the second row, the original images of the objects are given. In the third to the sixth rows, $\mathbf{f}^l_{hop}$ are visualized by displaying the part realizations with pixel intensity values $\|\mathbf{f}^l_{hop}\|^2_2$ for each $l = 1, 2, 3, 4$. $\mathbf{f}^l_{hog}$ features are visualized in the rest of the rows for each l.

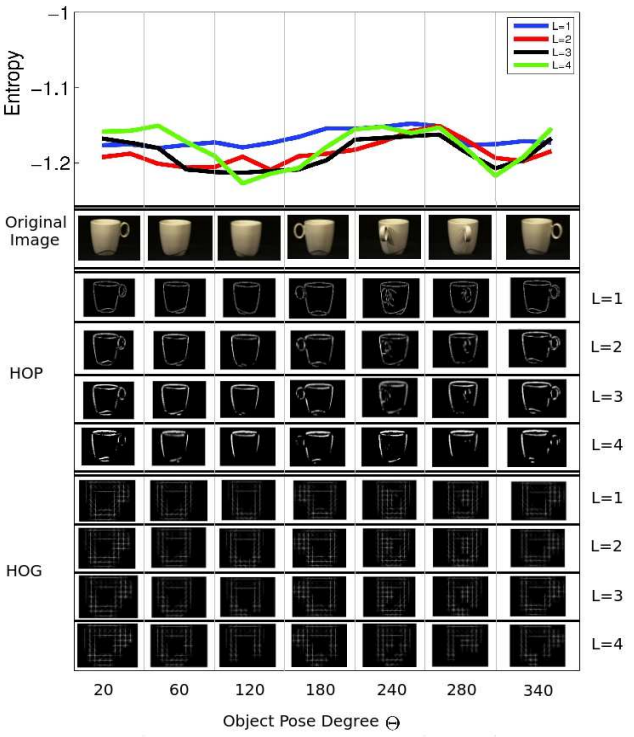


Fig. 3: Visualization of features extracted from part realizations for each different orientation of a cup and at each different layer of LHOP.

In Figure 3, we first observe that $f^{l=1}_{ent}$ values of the object change discriminatively across different object orientations Θ. For instance, if the handle of the cup is not seen from the front viewpoint of the cup (e.g. at $\Theta = 60°, 120°$), then we observe a smooth surface of the cup and the complexity of the part graphs, i.e. the entropy values, decrease. On the other

TABLE I: The samples that are selected from ALOI dataset and used in Object-wise Pose Estimation Experiments

Category Name	Apples	Balls	Bottles	Boxes	Cars	Cups	Shoes
Object IDs for Training	82	103	762	13	54	157	9
Object IDs for Testing	363, 540, 649, 710	164, 266, 291, 585	798, 829, 831, 965	110, 26, 46, 78	136, 138, 148, 158	36, 125, 153, 259	93, 113, 350, 826

hand, if the handle of the cup is observed at a front viewpoint (e.g. at $\Theta = 240°, 280°$), then the complexity increases. In addition, we observe that the difference between f^l_{ent} values of the object parts across different orientations Θ decreases as l increases. In other words, the discriminative power of the generative model of the LHOP increases at the higher layers of the LHOP since the LHOP captures the *important* parts and compositions that are co-occurred across different views through different layers.

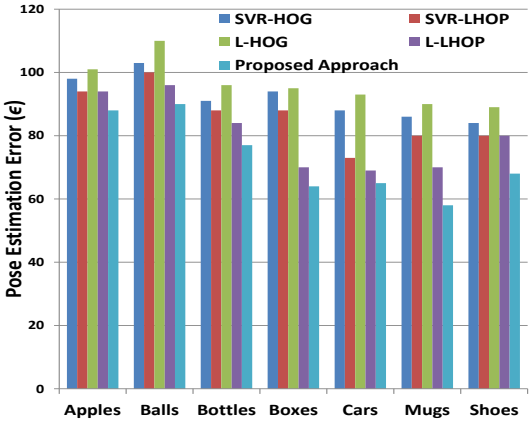


Fig. 4: Comparision of Object-wise Pose estimation errors (ϵ) of the proposed algorithms.

Given a ground truth Θ and an estimated pose value $\hat{\Theta}$, the pose estimation error is defined as $\epsilon = \|\Theta - \hat{\Theta}\|_2^2$. Pose estimation errors of state-of-the-art algorithms and the proposed Hierarchical Compositional Approach are given in Figure 4. In these results, we observe that the pose estimation errors of the algorithms which are implemented using the symmetric objects, such as apples and balls, are greater than that of the algorithms that are implemented on more structural objects such as cups.

In order to analyze this observation in detail, we show the ground truth Θ and the estimated orientations $\hat{\Theta}$ of some of the objects from Apples, Balls, cups and Shoes categories in Figure 5. We observe that some of the different views of the same object have the same shape and textural properties. For instance, the views of the ball at the orientations $\Theta = 10°$ and $\Theta = 225°$ represent the same pentagonal shape patterns. Therefore, similar parts are detected at these different views and the similar features are extracted from these detected parts. Then, the orientation of the ball, which is rotated by $\Theta = 10°$, is incorrectly estimated as $\hat{\Theta} = 225°$.

Fig. 5: Results for some of the objects from Apples, Balls, Cups and Shoes categories obtained in Object-wise Pose estimation experiments.

2) Experiments on Category-wise Pose Estimation: In Category-wise Pose Estimation experiments, we select different $\aleph^o_{tr}$ number of objects from different C number of categories as training images to estimate the pose of test objects, randomly. We employ the experiments on both ALOI and COIL datasets.

In the ALOI dataset, we randomly select $\aleph^o_{tr} = 1, 2, 3, 4$ number of training objects and $\aleph^o_{te} = 1$ test object which belong to Cups, Cow, Car, Clock and Duck categories. We repeat the random selection process two times and give the average pose estimation error for each experiment. In order to analyze the contribution of the information that can be obtained from the parts to the pose estimation performance using the part sharability of the LHOP, we initially select Cups and Cow categories ($C = 2$) and add new categories (Car, Clock and Duck) to the dataset, incrementally. The results are given in Table II. The results show that the pose estimation error decreases as the number of training samples, $\aleph^o_{tr}$, increases. This is due to the fact that the addition of new objects to the dataset increases the statistical representation capacity of the LHOP and the learning model of the regression algorithm. In addition, we observe that the pose estimation error observed in the experiments for $C = 2$ decreases when the objects from Car category are added to a dataset of objects belonging to Cups and Cow category in the experiments with $C = 3$. The performance boost is achieved by increasing the shareability of co-occurred object parts in different categories. For instance, the parts that construct the rectangular silhouettes of cows and cars can be shared in the construction of object representations in the LHOP (see Figure 6.

We employed two types of experiments on COIL dataset, constructing balanced and unbalanced training and test sets,

TABLE II: Category-wise Pose estimation errors (ϵ) of SVR-HOG/SVR-LHOP/L-HOG/L-LHOP/Proposed Approach for different number of categories (C) and training samples ($\aleph_{tr}^{o}$) selected from ALOI dataset.

$\aleph_{tr}^{o}$	C=2	C=3	C=4	C=5
1	133/103/140/97/91	116/99/110/97/89	110/95/102/95/88	102/94/99/95/88
2	130/100/133/95/85	108/93/104/88/81	105/91/95/88/80	100/94/100/91/85
3	105/91/104/86/75	93/83/87/83/70	99/86/94/84/75	95/81/93/75/70
4	94/86/90/73/68	90/79/84/73/65	92/77/86/72/64	95/75/88/71/60

Fig. 6: Sample images of the objects that are used in Category-wise Pose Estimation experiments.

in order to analyze the effect of the unbalanced data to the pose estimation performance. In the experiments, the objects are selected from Cat, Spatula, Cups and Car categories which contain 3, 3, 10 and 10 objects. Each object is rotated on a turntable by 5° from 0° to 355°.

In the experiments on balanced datasets, images of $\aleph_{tr}^{o}$ number of objects are initially selected from Cat and Spatula categories (for $C = 2$), and then images of the objects selected from Cups and Car categories are incrementally added to the dataset for $C = 3$ and $C = 4$ category experiments. More specifically, $\aleph_{tr}^{o}$ objects are randomly selected from each category and the random selection is repeated two times for each experiment. The results are shown in Table III. We observe that the addition of new objects to the datasets decreases the pose estimation error. Moreover, we observe a remarkable performance boost when the images of the objects from the categories that have similar silhouettes, such as Cat and Cups or Spatula and Car, are used in the same dataset.

TABLE III: Category-wise Pose estimation errors (ϵ) of SVR-HOG/SVR-LHOP/L-HOG/L-LHOP/Proposed Approach for different number of categories (C) and training samples ($\aleph_{tr}^{o}$) selected from COIL dataset.

$\aleph_{tr}^{o}$	C=2	C=3	C=4
1	125/109/120/95/85	120/85/103/77/68	110/79/95/71/62
2	120/95/114/89/77	93/77/81/63/59	104/76/92/69/51

We prepared unbalanced datasets by randomly selecting the images of $\aleph_{te}^{o} = 1$ object from each category as a test sample and the images of the rest of the objects belonging to the associated category in the COIL dataset as training samples. For instance, the images of a randomly selected cat are selected as test samples and the images of the remaining two cats are selected as training samples. This procedure is repeated two times in each experiment and the average values of pose estimation errors are depicted in Figure 7. The results show that SVR is more sensitive to the balance of the dataset and the number of training samples than the proposed approach. For instance, the difference between the pose estimation error of SVR given in Table III and Figure 7 for $C = 4$ is approximately 10°, while that of the proposed Hierarchical Compositional Approach is approximately 5°.

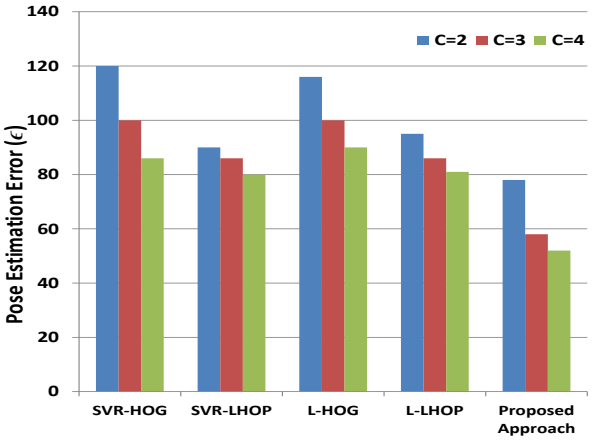


Fig. 7: Category-wise Pose estimation errors (ϵ) of the state-of-the-art algorithms and the proposed Hierarchical Compositional Approach in the experiments on COIL dataset.

In the next subsection, the experiments on object categorization are given.

B. Experiments on Object Categorization

In the Object Categorization experiments, we use the same experimental settings that are described in Section V-A.2 for Category-wise Pose Estimation.

TABLE V: Categorization performance (%) of SVM-HOG/SVM-LHOP/LR-HOG/LR-LHOP/Proposed Approach using COIL dataset.

$\aleph_{tr}^{o}$	C=2	C=3	C=4
1	94/93/92/95/100	89/88/91/91/97	81/79/80/81/84
2	97/97/96/97/100	89/91/90/93/97	84/86/83/87/90

The results of the experiments employed on ALOI dataset and balanced subsets of COIL dataset are given in Table IV and Table V, respectively. In these experiments, we observe that the categorization performance decreases as the number of categories increases. However, we observe that the pose estimation error decreases as the number of

TABLE IV: Categorization performance (%) of SVM-HOG/SVM-LHOP/LR-HOG/LR-LHOP/Proposed Approach for different number of categories (C) and training samples ($\aleph_{tr}^{o}$) selected from ALOI dataset.

$\aleph_{tr}^{o}$	C=2	C=3	C=4	C=5
1	88/89/91/93/100	85/88/84/92/98	85/85/84/85/90	81/81/81/83/90
2	88/91/92/94/100	88/91/87/93/98	87/87/86/88/92	81/83/81/84/91
3	95/98/94/98/100	91/93/91/95/99	90/90/90/91/93	83/85/83/88/91
4	97/98/98/99/100	93/96/93/97/100	90/91/90/91/94	87/91/89/95/96

categories increases in the previous sections. The reason of the observation of this error difference is that the objects rotated on a turn table may provide similar silhouettes although they may belong to different categories. Therefore, addition of the images of new objects that belong to different categories may boost pose estimation performance. On the other hand, addition of the images of these new objects may decrease the categorization performance if the parts of the object cannot be shared across different categories and increase the data complexity of the feature space.

VI. Conclusion

In this paper, we have proposed a compositional hierarchical approach for joint object pose estimation and categorization using a generative-discriminative learning method. The proposed approach first exposes information about pose and category of an object by extracting features from its realizations observed at different layers of LHOP in order to consider different levels of abstraction of information represented in the hierarchy. Next, we formulate joint object pose estimation and categorization problem as a sparse optimization problem. Then, we solve the optimization problem by integrating the features extracted at each different layer using a distributed and parallel optimization algorithm.

We examine the proposed approach on benchmark 2-D multi-view image datasets. In the experiments, the proposed approach outperforms state-of-the-art Support Vector Machines for object categorization and Support Vector Regression algorithm for object pose estimation. In addition, we observe that shareability of object parts across different object categories and views may increase pose estimation performance. On the other hand, object categorization performance may decrease as the number of categories increases if parts of an object cannot be shared across different categories, and increase the data complexity of the feature space. The proposed approach can successfully estimate the pose of objects which have view-specific statistical and geometric properties. On the other hand, the proposed feature extraction algorithms cannot provide information about the view-specific properties of symmetric or semi-symmetric objects, which leads to a decrease of the object pose estimation and categorization performance. Therefore, the ongoing work is directed towards alleviating the problems with symmetric or semi-symmetric objects.

References

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[2] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, May 2011.

[3] A. Collet, D. Berenson, S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *Proc. IEEE Conf. Robotics and Automation*, 2009, pp. 48–55.

[4] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," *Int. J. Rob. Res.*, vol. 30, no. 10, pp. 1284–1306, Sep 2011.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.

[6] W. Du, X. Li, Y. Li, and S. Severini, "A note on the von neumann entropy of random graphs." *Linear Algebra Appl.*, vol. 433, no. 11-12, pp. 1722–1725, 2010.

[7] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[8] S. Fidler, M. Boben, and A. Leonardis, *Object Categorization: Computer and Human Vision Perspectives*. Cambridge University Press, 2009, ch. Learning Hierarchical Compositional Representations of Object Structure.

[9] ——, "A coarse-to-fine taxonomy of constellations for fast multi-class object detection," in *Proceedings of the 11th European Conference on Computer Vision: Part V*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 687–700.

[10] J.-M. Geusebroek, G. Burghouts, and A. Smeulders, "The amsterdam library of object images," *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 103–112, 2005.

[11] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and continuous pose estimation," *Image Vision Comput*, vol. 30, pp. 923–933, 2012.

[12] T. Haštie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer-Verlag, 2001.

[13] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *Int. J. Rob. Res.*, vol. 31, no. 9, pp. 1021–1043, Aug 2012.

[14] G. Kootstra, M. Popović, J. A. Jørgensen, K. Kuklinski, K. Miatliuk, D. Kragic, and N. Krüger, "Enabling grasping of unknown objects through a synergistic use of edge and surface information," *Int. J. Rob. Res.*, vol. 31, no. 10, pp. 1190–1213, Sep 2012.

[15] K. Lai, L. Bo, X. Ren, and D. Fox, "A scalable tree-based approach for joint object and pose recognition," in *Proc. The 25th AAAI Conf. Artificial Intelligence*, Aug 2011.

[16] A. Mowshowitz and M. Dehmer, "Entropy and the complexity of graphs revisited." *Entropy*, vol. 14, no. 3, pp. 559–570, 2012.

[17] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-100)," Department of Computer Science, Columbia University, Tech. Rep., Feb 1996.

[18] M. Ozay, I. Esnaola, F. Vural, S. Kulkarni, and H. Poor, "Sparse attack construction and state estimation in the smart grid: Centralized and distributed models," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 7, pp. 1306–1318, 2013.

[19] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J Comput Graph Stat*, vol. 10, pp. 231–245, 2012.

[20] D. Teney and J. Piater, "Probabilistic object models for pose estimation in 2d images," in *Pattern Recognition*, ser. Lecture Notes in Computer Science, R. Mester and M. Felsberg, Eds. Springer Berlin Heidelberg, 2011, vol. 6835, pp. 336–345.

A Graph Theoretic Approach for Object Shape Representation in Compositional Hierarchies using a Hybrid Generative-Descriptive Model *

Umit Rusen Aktas, Mete Ozay, Aleš Leonardis and Jeremy Wyatt

School of Computer Science, The University of Birmingham, Edgbaston, Birmingham, B15 2TT, United Kingdom.
Emails: { xxa334, m.ozay, a.Leonardis, j.l.wyatt } @cs.bham.ac.uk

Abstract. A graph theoretical approach is proposed for object shape representation in a hierarchical compositional architecture called Compositional Hierarchy of Parts (CHOP). In the proposed approach, vocabulary learning is performed using a hybrid generative-descriptive model. Two information theoretic algorithms are used for learning a vocabulary of compositional parts. First, statistical relationships between parts are quantified as the amount of information needed to describe a realization of a shape part given the realizations of other parts on 2D images. The statistical relationships are learned using a *Minimum Conditional Entropy Clustering* algorithm. Second *contribution* of a part to representation of a shape in a part composition is *described* by measuring a *conditional description length* of the part given a compositional representation of the shape at a layer of the hierarchy. Then, part selection problem is defined as a Subgraph Isomorphism problem, and solved using an MDL principle. Finally, part compositions are constructed considering learned statistical relationships between parts and their conditional description length.
The proposed approach and algorithms are examined using a multiple view image dataset and two articulated image datasets. Experimental results show that CHOP can recognize and use part shareability property in the construction of vocabularies and inference trees. For instance, if parts of shapes encoded in a learned vocabulary and a new given shape, which will be used for incremental learning of vocabulary, are shareable, then the shareable parts can be used to improve the statistical relationships between learned parts, and minimize description length of parts and compositions in the CHOP. Additionally, junctions and closed curves observed at the shape boundaries can be detected as part realizations if they are shared among different articulated images.

1 Introduction

Hierarchical compositional architectures have been studied in the literature for representation of object categories [7,13,14], face reconstruction [16], object de-

* An extended version of the report is submitted to The European Conference on Computer Vision, 2014.

tection [5] and parsing [17]. A detailed review of the recent works is given in [18], and the relationship between hierarchical compositional architectures and deep learning algorithms for learning representations is analyzed in [1].

In this paper, we consider a hierarchical compositional architecture for the representation and recognition of shapes in two dimensional images. In [11] and [6], shape models are learned using hierarchical shape matching algorithms. Kokkinos and Yuille [10] first decompose object categories into parts and shape contours using a top-down approach. Then they employ a Multiple Instance Learning algorithm to discriminatively learn the shape models using a bottom-up approach. However, part-shareability and indexing mechanisms [8] are not employed and considered as future work in [10].

Fidler, Boben and Leonardis [8] analyzed crucial properties of hierarchical compositional approaches that should be invoked by the proposed architectures. Following their analyses, we study on unsupervised generative bottom-up learning of vocabulary of parts considering part-shareability, and performing *efficient* inference of object shapes on test images using an indexing and matching method.

The work most related to the proposed hierarchical architecture is *Learned Hierarchy of Parts* (LHOP) proposed by Fidler and Leonardis [7]. At the first layer of the LHOP, first Gabor filters are employed on the images to obtain Gabor features, which are defined as *first layer parts*. Next, the statistical properties of the distributions of *part realizations* are learned by first using a *local inhibition* method to reduce redundancy of the representation of neighboring parts, and then computing the frequent co-occurrences of part types and their relative locations [8]. Then, the part compositions that will be constructed at the next layer are *inferred* using an inference algorithm based on Expectation Maximization (EM) and Markov Chain Monte Carlo (MCMC) methods [7]. These processes are recursively employed at each layer to construct a hierarchical vocabulary of parts and their compositions [7,8].

In this paper, we employ two information theoretic methods to learn the statistical properties of parts, and construct the compositions of parts by minimizing their description length. First we model the relationship between parts using a Minimum Conditional Entropy Clustering algorithm [12], in order to construct compositions of varying number of parts, i.e. compositions of C-parts instead of two-parts called duplets [7,8], where C is the number of clusters which represent the conditional distributions of pairwise parts in local spatial neighborhoods in the images. Second, we define part descriptions as graphs at a layer l of the hierarchy. Then we infer the compositions of parts at the consecutive layers $l+1$, $\forall l = 1, 2, \ldots, L$ of an L-layer compositional hierarchy by computing a subgraph of an ensemble of part graphs, and minimizing their description length. Minimum Description Length (MDL) models have been employed for statistical shape analysis [4,15] to achieve compactness, specificity and generalization ability properties of shape models [4].

Our contributions in this work are threefold:

1. We introduce a graph theoretical approach to represent objects and parts in compositional hierarchies. Although other hierarchical methods also use graphs as data structures [5,10,17], to our knowledge, CHOP is the first system to fully encode and infer compositional parts of objects using hybrid generative-descriptive learning models within a graph-based hierarchical compositional framework to the best of our knowledge. Additionally, the proposed approach enables us to use graph theoretical tools to analyze, measure and employ geometric and statistical properties of parts to construct part compositions.
2. Two information theoretic methods are used in the proposed CHOP algorithm. For this purpose, we define parts as random graphs and represent part realizations as the instances of random graphs observed on images in datasets. First we learn statistical relationships between parts using a *Minimum Conditional Entropy Clustering* algorithm. Then, we compute the statistical relationship between two parts by measuring the amount of information needed to describe the part realization R_i of a part P_i given that the part realization R_j of another part P_j, for all parts P_i, P_j represented in a learned vocabulary, and for all realizations R_i, R_j observed on images. Second we define *contribution* of a part P_i to representation of a shape in a part composition by measuring *conditional description length* of P_i given a compositional representation of the shape at a layer of the hierarchy using an MDL principle. In order to select the parts which represent compositional shapes with minimum description lengths, we solve a Subgraph Isomorphism problem. Finally, part compositions are constructed considering learned statistical relationships between parts and their description length.
3. CHOP employs a hybrid generative-descriptive model for hierarhical compositional representation of shapes. The proposed model differs from other frequency-based approaches in that the part selection process is driven by the MDL principle, which effectively selects parts which are not only frequently observed, but also provide *descriptive* information for the representation of shapes.

The paper is organized as follows. The proposed Compositional Hierarchy of Parts (CHOP) algorithm is given in the next section. Preprocessing step is explained in Section 2.1. In Section 2.2, statistical learning and inference algorithms used for the construction of vocabularies are given. An algorithm used for inference of object shapes on test images is described in Section 2.3. Experimental analyses are given in Section 3, and Section 4 concludes the paper.

2 Compositional Hierarchy of Parts

In this section, we give the descriptions of the algorithms employed in the proposed Compositional Hierarchy of Parts (CHOP) in training and testing phases. In the next section, we first describe the preprocessing algorithms that are used in both training and testing phases. Next, we introduce the vocabulary learning algorithms in Section 2.2. Then, we describe the inference algorithms performed

on the test images for the representation of object shapes and categories in Section 2.3.

2.1 Preprocessing

Given a set of images $S = \{s_n, y_n\}_{n=1}^{N}$, where $y_n \in \mathbb{Z}^+$ is the category label of an image s_n, we first extract a set of Gabor features $F_n = \{f_{nm}(\mathbf{x}_{nm}) \in \mathbb{R}\}_{m=1}^{M}$ from each image s_n using Gabor filters employed at location $\mathbf{x}_{nm}$ in s_n at Θ orientations, where

$$f_{nm}(\mathbf{x}_{nm}) = \underset{f_{nm}(\mathbf{x}_{nm},\theta)}{\arg\max} \{f_{nm}(\mathbf{x}_{nm}, \theta)\}_{\theta=1}^{\Theta}.$$

Then, we construct a set of Gabor features $F = \bigcup_{n=1}^{N} F_n$. In this work, we compute the Gabor features at $\Theta = 6$ different orientations.

In order to remove the redundancy of Gabor features in the images, we perform a non-maxima suppression. In this step, a Gabor feature with the Gabor response value $f_{nm}(\mathbf{x}_{nm})$ is removed from F_n if $f_{nm}(\mathbf{x}_{nm}) < f_{na}(\mathbf{x}_{na})$, for all Gabor features extracted at $\mathbf{x}_{na} \in \aleph(\mathbf{x}_{nm})$, where $\aleph(\mathbf{x}_{nm})$ is a set of image positions of the Gabor features that reside in the neighborhood of $\mathbf{x}_{nm}$ defined by Euclidean distance in $\mathbb{R}^2$. After inhibition is performed, we obtain a set of suppressed Gabor features $\hat{F}_n \subset F_n$ and $\hat{F} = \bigcup_{n=1}^{N} \hat{F}_n$.

In this section, we assume that the set of images S is split into two non-overlapping training and test sets of images, such that $S^{tr} \cup S^{te} = S$ and $S^{tr} \cap S^{te} = \varnothing$, $N_{te} = |S^{te}|$, $N_{tr} = |S^{tr}|$ and $|\cdot|$ represents the cardinality of the set. S^{tr} is only used for learning vocabulary of parts, and S^{te} is only used for inference of the representation of object shapes and categories in the testing phase. In other words, S^{te} is not available to the vocabulary learning algorithm and S^{tr} is not used in testing.

2.2 Learning Vocabulary of Parts

Given a set of training images S^{tr}, we first learn the statistical properties of parts using their realizations on images at a layer l in the CHOP. Then, we infer the compositions of parts that will be constructed at layer $l + 1$ by minimizing the description length of the part descriptions defined as *Object Graphs*. In order to remove the redundancy of the compositions, we employ a *local inhibition* process that is suggested by Fidler and Leonardis [7]. Statistical learning of part structures, inference of compositions and local inhibition processes are performed by recursively constructing parts and their compositions at each layer, and the details of the algorithms are given in the following subsections.

Definitions In this section, we define parts, part realizations and graph structures used in the CHOP. We first define parts and part realizations below.

Definition 1 (Parts and Part Realizations).

The i^{th} part constructed at the l^{th} layer $\mathcal{P}_i^l = (\mathcal{G}_i^l, \mathcal{Y}_i^l)$ is a tuple consisting of a directed random graph $\mathcal{G}_i^l = \{\mathcal{V}_i^l, \mathcal{E}_i^l\}$, where $\mathcal{V}_i^l$ is a set of nodes and $\mathcal{E}_i^l$ is a set of edges, and a random variable $\mathcal{Y}_i^l \in \mathbb{Z}^+$ which represents the identity number or label of the part.

The realization $R_i^l(s_n) = (G_i^l(s_n), Y_i^l(s_n))$ of $\mathcal{P}_i^l$ is defined by 1) $Y_i^l(s_n)$ which is the realization of $\mathcal{Y}_i^l$ representing the label of the part realization on an image (s_n), and 2) the directed graph $G_i^l(s_n) = \{V_i^l(s_n), E_i^l(s_n)\}$ which is an instance of the random graph $\mathcal{G}_i^l$ computed on an training image $(s_n) \in S^{tr}$, where $V_i^l(s_n)$ is a set of nodes and $E_i^l(s_n)$ is a set of edges of $G_i^l(s_n)$, $\forall n = 1, 2, \ldots, N_{tr}$.

At the first layer $l = 1$, each node of $\mathcal{V}_i^1$ is a part label $\mathcal{Y}_i^1 \in \mathcal{V}_i^1$ taking values from the set $\{1, 2, \ldots, \Theta\}$, and $\mathcal{E}_i^1 = \varnothing$. Similarly, $E_i^1(s_n) = \varnothing$, and each node of $V_i^1(s_n)$ is defined as a Gabor feature $f_{na}^i(\mathbf{x}_{na}) \in \hat{F}_n^{tr}$ observed in the image $s_n \in S^{tr}$ at the image location $\mathbf{x}_{na}$, i.e. the a^{th} realization of $\mathcal{P}_i^l$ observed in $s_n \in S^{tr}$ at $\mathbf{x}_{na}$, $\forall n = 1, 2, \ldots, N_{tr}$.

In the consecutive layers, the parts and part realizations are defined recursively by employing layer-wise mappings $\Psi_{l,l+1}$ as

$$\Psi_{l,l+1} : (\mathcal{P}^l, R^l, \mathbb{G}_l) \rightarrow (\mathcal{P}^{l+1}, R^{l+1}), \forall l = 1, 2, \ldots, L, \tag{1}$$

where $\mathcal{P}^l = \{\mathcal{P}_i^l\}_{i=1}^{A_l}$, $R^l = \{R_i^l(s_n) : \forall s_n \in S^{tr}\}_{i=1}^{B_l}$, $\mathcal{P}^{l+1} = \{\mathcal{P}_j^{l+1}\}_{j=1}^{A_{l+1}}$, $R^{l+1} = \{R_j^{l+1}(s_n) : \forall s_n \in S^{tr}\}_{j=1}^{B_{l+1}}$ and $\mathbb{G}_l$ is an object graph which is defined next. □

In the rest of this section, we will use $R_j^l(s_n) \triangleq R_j^l$, $\forall i = 1, 2, \ldots, B_l$, $\forall l = 1, 2, \ldots, L$, $\forall s_n \in S^{tr}$, for the sake of simplicity in the notation.

Definition 2 (Receptive Field).

A receptive field of a part realization $\mathcal{R}_i^l$ is an acyclic and tree-shaped graph $RF_i^l = (V_i^l, E_i^l)$, with root node being $\mathcal{R}_i^l$. A directed edge $e_{ab} \in E_i^l$ is defined as

$$e_{ab} = \begin{cases} (a^l, b^l, \phi_{ab}^l), & \text{if} \quad \mathbf{x}_{nb} \in \aleph(\mathbf{x}_{na}), a = i \\ \varnothing, & otherwise \end{cases}, \tag{2}$$

where $\aleph(\mathbf{x}_{na})$ is the set of part realizations that reside in a neighborhood of a part realization R_a^l in an image s_n, $\forall R_a^l, R_b^l \in V_i^l, b \neq i$ and $\forall s_n \in S^{tr}$.

ϕ_{ab}^l defines the statistical relationship between R_a^l and R_b^l, and is computed as described in the next subsection.

Definition 3 (Object Graph).

Structure of part realizations observed at the l^{th} layer on the training set S^{tr} is described using a directed graph $\mathbb{G}_l = (\mathbb{V}_l, \mathbb{E}_l)$, called an object graph, where $\mathbb{V}_l = \bigcup_i V_i^l$ is a set of nodes, where $V_i \in RF_i$, $\forall i$, and $\mathbb{E}_l = \bigcup_i E_i^l$ is a set of edges, $E_i \in RF_i$, $\forall i$.

Learning of Statistical Relationships between Parts and Part Realizations Statistical relationships between parts and their realizations are learned using S^{tr} using two approaches.

In the first approach, we compute the *conditional* distributions $P_{\mathcal{P}_i^l}(R_a^l|\mathcal{P}_j^l = R_b^l)$, $i = Y_a^l$ and $j = Y_b^l$ between all possible pairs of parts $(\mathcal{P}_i^l, \mathcal{P}_j^l)$ using S^{tr} at the l^{th} layer. However, we select a set of modes $\mathcal{M}^l = \{M_{ij} : i = 1, 2, \ldots, B_l, j = 1, 2, \ldots, B_l\}$, where $M_{ij} = \{M_{ijk}\}_{k=1}^K$ of these distributions instead of detecting a single mode. For this purpose, we define the mode computation problem as a *Minimum Conditional Entropy Clustering* problem [12] as

$$Z_{ijk} \coloneqq \arg\min_{k \in C} - \sum_{\forall \mathbf{x}_{na}^l} \sum_{k=1}^{K} P(k, R_a^l|R_b^l) \log P(k, R_a^l|R_b^l),$$

where the first summation is over all part realizations R_a^l that reside in a neighborhood of all R_b^l such that $\mathbf{x}_{na}^l \in \aleph(\mathbf{x}_{nb}^l)$, for all $i = Y_a^l$ and $j = Y_b^l$, C is a set of cluster ids, $K = |C|$ is the number of clusters, $k \in C$ is a cluster label, and $P(k, R_a^l|R_b^l) \triangleq P_{\mathcal{P}_i^l}(k, R_a^l|\mathcal{P}_j^l = R_b^l)$.

The pairwise statistical relationship between two part realizations R_a^l and R_b^l is represented as $M_{ijk} = (i, j, \mathbf{c}_{ijk}, Z_{ijk})$, where $\mathbf{c}_{ijk}$ is the center position of the k^{th} cluster. In the construction of an object graph $\mathbb{G}_l$ at the l^{th} layer, we compute $\phi_{ab}^l = (\mathbf{c}_{ijk}, \hat{k})$, $\forall a, b$ as

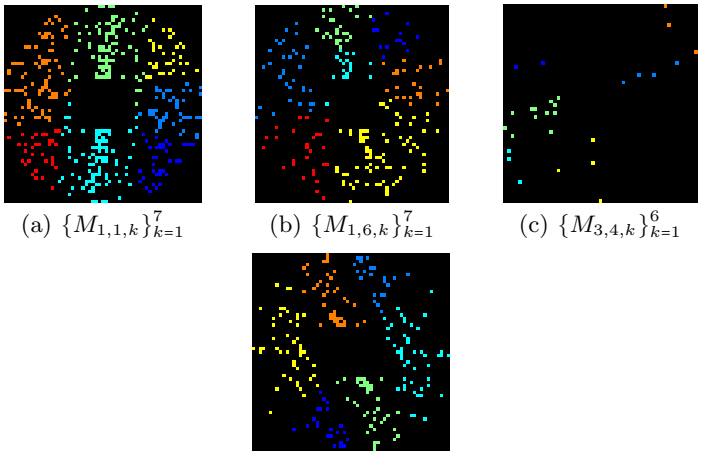
$$\hat{k} = \arg\min_{k \in C} \|\mathbf{d}_{ab} - \mathbf{c}_{ijk}\|_2,$$

where $\|\cdot\|_2$ is the Euclidan distance, $i = Y_a^l$ and $j = Y_b^l$, $\mathbf{d}_{ab} = \mathbf{x}_{na} - \mathbf{x}_{nb}$, $\mathbf{x}_{na}$ and $\mathbf{x}_{na}$ are the positions of R_a^l and R_j^l in an image s_n, respectively.

In Fig. 1, clustered samples which are used to calculate conditional distributions $P_{\mathcal{P}_i^1}(R_i^l|\mathcal{P}_j^1 = R_j^1)$ at layer $l = 1$ are illustrated. Each cluster corresponds to a mode M_{ijk} in $\mathcal{M}^1 = \{M_{ij} : i, j = 1, 2, \ldots, B_1\}$, where $\mathbf{c}_{ijk}$ is calculated as the center of k^{th} cluster using distribution of relative part realization positions of $\mathcal{P}_i$ and $\mathcal{P}_j$.

In the second approach, we employ fixed-size bins to partition all possible configurations, effectively discretizing 2-D Euclidean space. The partitioning used in experiments is shown in Fig. 2. For this specific setup, there are 8 modes $\{M_{i,j,k}\}_{k=1}^8$, $\forall i, j$.

Inference of Compositions of Parts using MDL Given a set of parts $\mathcal{P}^l$, a set of part realizations $\mathcal{R}^l$, and an object graph $\mathbb{G}_l$ at the l^{th} layer, we infer compositions of parts at the $(l+1)^{st}$ layer by computing a mapping $\Psi_{l,l+1}$ in (1). In this mapping, we search for a structure which *best describes* structure of parts $\mathcal{P}^l$ as the compositions constructed at the $(l+1)^{st}$ layer by minimizing the length of description of $\mathcal{P}^l$. In the inference process, we search a set of graphs $\mathcal{G}^{l+1} = \{\mathcal{G}_j^{l+1}\}_{j=1}^{B_{l+1}}$ which minimizes the description length of $\mathbb{G}_l$ as



(a) $\{M_{1,1,k}\}_{k=1}^{7}$ (b) $\{M_{1,6,k}\}_{k=1}^{7}$ (c) $\{M_{3,4,k}\}_{k=1}^{6}$

(d) $\{M_{6,6,k}\}_{k=1}^{6}$

Fig. 1: Example conditional distributions in $\mathcal{M}^1$. (Best viewed in colour)

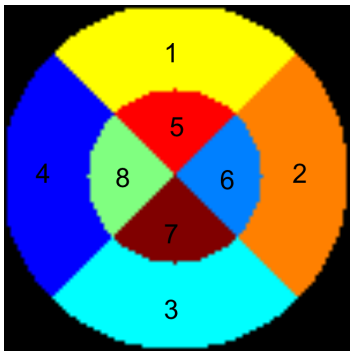


Fig. 2: Bin-based partitioning of relative configurations.

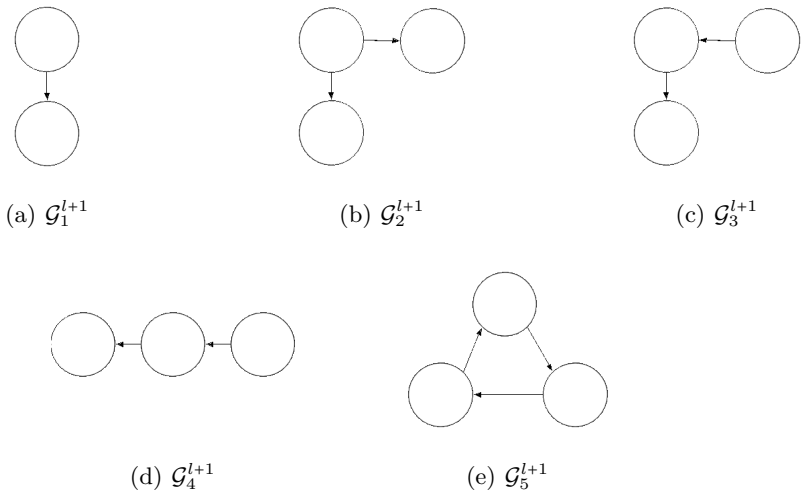
$$\mathcal{G}^{l+1} = \underset{\{\mathcal{G}_j^{l+1}: j=1,2,\ldots,B_{l+1}\}}{\arg\min} \; value(\mathcal{G}_j^{l+1}, \mathbb{G}_l), \tag{3}$$

where

$$value(\mathcal{G}_j^{l+1}, \mathbb{G}_l) = \frac{DL(\mathcal{G}_j^{l+1}) + DL(\mathbb{G}_l|\mathcal{G}_j^{l+1})}{DL(\mathbb{G}_l)} \tag{4}$$

is the compression value of an object graph $\mathbb{G}_l$ given a subgraph $\mathcal{G}_j^{l+1}$ of a receptive field G_j^{l+1}, $\forall j = 1, 2, \ldots, B_{l+1}$. This unsupervised part discovery process consists of two steps:

1. **Enumeration:** In graph enumeration step, candidate graphs $\mathcal{G}^{l+1}$ are generated from $\mathbb{G}_l$. However, each $\mathcal{G}_j^{l+1} \in \mathbb{G}_l$ is forced to include nodes $\mathcal{V}_j^{l+1}$ and edges $\mathcal{E}_j^{l+1}$ from only one receptive field RF_i^l, $\forall i$. In effect, this selective candidate generation procedure enforces $\mathcal{G}_j^{l+1}$ to represent an area around its center node. Examples of valid and invalid candidates are illustrated in Fig. 3. While $\mathcal{G}_1^{l+1}$ and $\mathcal{G}_2^{l+1}$ are valid structures, $\mathcal{G}_3^{l+1}$, $\mathcal{G}_4^{l+1}$, and $\mathcal{G}_5^{l+1}$ are not enumerated since they have nodes/edges received from different receptive fields.



(a) $\mathcal{G}_1^{l+1}$ (b) $\mathcal{G}_2^{l+1}$ (c) $\mathcal{G}_3^{l+1}$

(d) $\mathcal{G}_4^{l+1}$ (e) $\mathcal{G}_5^{l+1}$

Fig. 3: Valid and invalid candidates.

2. **Evaluation:** Once we obtain $\mathcal{G}^{l+1}$ by solving (3) with $\mathcal{G}^{l+1}$ subject to constraints provided in the previous step, we compute a set of graph instances of part realizations $G^{l+1} = \{G_j^{l+1}\}_{j=1}^{B_{l+1}}$ such that $G_j^{l+1} \in iso(\mathcal{G}_j^{l+1})$ and $G_j^{l+1} \subseteq \mathbb{G}_l$, where $iso(\mathcal{G}_j^{l+1})$ is a set of all graphs that are isomorphic to $\mathcal{G}_j^{l+1}$. This problem is defined as a subgraph isomorphism problem [3], which is NP-complete.

In this work, the proposed graph structures are acyclic and tree-shaped, enabling us to solve the problem (3) in polynomial time. In order to obtain two sets of subgraphs $\mathcal{G}^{l+1}$ and G^{l+1} by solving (3), we have implemented a simplified version of the substructure discovery system, SUBDUE [3] which uses a restricted search space as explained above.

The label of a part $\mathcal{P}_j^{l+1}$ is defined according to its compression value

$$\mu_j^{l+1} \triangleq value(\mathcal{G}_j^{l+1}, \mathbb{G}_l)$$

computed in (4). We sort compression values in a descending order $\mu_1^{l+1} \geq \mu_2^{l+1} \geq \ldots \geq \mu_{A_{l+1}}^{l+1}$ to construct a list of ordered labels $\mu_{(1)}^{l+1} \geq \mu_{(2)}^{l+1} \geq \ldots \geq \mu_{(A_{l+1})}^{l+1}$, such that

$$\mu_{(1)}^{l+1} = \arg\max_{\mu_j^{l+1}} \{\mu_j^{l+1}\}_{j=1}^{A_{l+1}},\ \mu_{(A_{l+1})}^{l+1} = \arg\min_{\mu_j^{l+1}} \{\mu_j^{l+1}\}_{j=1}^{A_{l+1}}, \tag{5}$$

and $\mu_{(k)}^{l+1}$ is the k^{th} maximum compression value. Then, the label of a part $\mathcal{P}_j^{l+1}$ with $\mu_{(k)}^{l+1}$ is $\mathcal{Y}_j^{l+1} = k$.

After sets of graphs and part labels are constructed at the $(l+1)^{st}$ layer, we construct a set of parts $\mathcal{P}^{l+1} = \{\mathcal{P}_i^{l+1}\}_{i=1}^{A_{l+1}}$, where $\mathcal{P}_i^{l+1} = (G_i^{l+1}, \mathcal{Y}_i^{l+1})$. We call $\mathcal{P}^{l+1}$ a set of *compositions* of the parts in the set $\mathcal{P}^l$ constructed at the $(l+1)^{st}$ layer. Similarly, we construct a set of part realizations $\hat{R}^{l+1} = \{R_j^{l+1}\}_{j=1}^{B_{l+1}}$, where $R_j^{l+1} = (G_j^{l+1}, Y_j^{l+1})$.

In order to remove the redundancy in the set of part realizations, we perform local inhibition as suggested in [7] and obtain a new set of part realizations $R^{l+1} \subseteq \hat{R}^{l+1}$.

Incremental Construction of the Vocabulary We define the vocabulary of the CHOP below.

Definition 4 (Vocabulary). *A tuple $\Omega_l = (\mathcal{P}^l, \mathcal{M}^l)$ is the vocabulary constructed at the l^{th} layer using the training set S^{tr}. The vocabulary of a CHOP with L layers is defined as the set $\Omega = \{\Omega_l : l = 1, 2, \ldots, L\}$.* □

We construct Ω of CHOP incrementally as described in the pseudo-code of vocabulary learning algorithm given in Algorithm 1. Given a set of training images $S^{tr} = \{s_n\}_{n=1}^{N^{tr}}$, and the number of orientations of Gabor features D, we first pre-process the training images to construct parts and their realization at the first layer $l = 1$. In the first step of the algorithm, we extract a set of Gabor features $F_n = \{f_{nm}(\mathbf{x}_{nm})\}_{m=1}^{M}$ from each image $s_n \in S^{tr}$ using Gabor filters employed at location $\mathbf{x}_{nm}$ in s_n at D orientations. Then, we perform local inhibition of Gabor Features using non-maxima suppression to construct a set of suppressed Gabor features $\hat{F}_n \subset F_n$ as described in Section 2.1 in the second step. Next, we initialize the variable l which defines the layer index, and we construct parts $\mathcal{P}^1$ and part realizations R^1 at the first layer as described in Definition 1. Before processing the new layer, images are subsampled by changing the scale of

part realizations R^l, which effectively increases the area receptive fields through upper layers.

In steps **5 – 11**, we incrementally construct the vocabulary of the CHOP. In step **5**, we compute the sets of modes $\mathcal{M}^l$ by learning statistical relationships between part realizations as described in Section 2.2. In the sixth step, we construct an object graph $\mathbb{G}_l$ using $\mathcal{M}^l$ as explained in Definition 3, and we construct the vocabulary $\Omega_l = (\mathcal{P}^l, \mathcal{M}^l)$ at the l^{th} layer in step **7**. Next, we infer part graphs that will be constructed at the next layer $\mathcal{G}_{l+1}$ by computing the mapping $\Psi_{l,l+1}$. For this purpose, we solve (3) using our graph mining implementation to obtain a set of parts $\mathcal{P}^{l+1}$ and a set of part realizations R^{l+1} as explained in Section 2.2. We increment l in step **10**, and subsample the positions of part realizations R_i^l by a factor of σ, $\forall n, R_i^l$ in step **11**, and iterate the steps **5 – 11** while a non-empty part graph $\mathcal{G}^l$ is either obtained from the training images at the first layer, or inferred from Ω_{l-1}, R_{l-1} and $\mathbb{G}_{l-1}$ at $l > 1$, i.e. $\mathcal{G}^l \neq \varnothing$, $\forall l \geq 1$. At the output of the algorithm, we obtain the vocabulary of CHOP, $\Omega = \{\Omega_l : l = 1, 2, \ldots, L\}$.

Input :
- $S^{tr} = \{s_n\}_{n=1}^{N^{tr}}$: Training dataset,
- Θ: The number of different orientations of Gabor features,
- σ: Subsampling ratio.

Output: Vocabulary Ω.

1 Extract a set of Gabor features $F^{tr} = \bigcup_{n=1}^{N} F_n^{tr}$, where $F_n^{tr} = \{f_{nm}(\mathbf{x}_{nm})\}_{m=1}^{M}$ from each image $s_n \in S^{tr}$;
2 Construct a set of suppressed Gabor features $\hat{F}^{tr} \subset F^{tr}$ (see Section 2.1);
3 $l := 1$;
4 Construct $\mathcal{P}^1$ and R^1 (see Definition 1);
while $\mathcal{G}^l \neq \varnothing$ **do**
5 Compute the sets of modes $\mathcal{M}^l$ (see Section 2.2);
6 Construct $\mathbb{G}_l$ using $\mathcal{M}^l$ (see Definition 3);
7 Construct $\Omega_l = (\mathcal{P}^l, \mathcal{M}^l)$;
8 Infer part graphs $\mathcal{G}_{l+1}$ by solving (3) (see Section 2.2);
9 Construct $\mathcal{P}^{l+1}$ and R^{l+1} (see Section 2.2);
10 $l := l + 1$;
11 Subsample the positions of part realizations R_i^l by a factor of σ, $\forall n, R_i^l$;
end
12 $\Omega = \{\Omega_t : t = 1, 2, \ldots, l-1\}$;

Algorithm 1: Vocabulary learning algorithm of Compositional Hierarchy of Parts.

2.3 Inference of Object Shapes on Test Images

In the testing phase, we infer shapes of objects on test images $s_n \in S^{te}$ using the learned vocabulary of parts Ω.

We incrementally construct a set of inference graphs $\mathcal{T}(s_n) = \{\mathcal{T}(s_n)\}_{l=1}^{L}$ of a given test image $s_n \in S^{te}$ using the learned vocabulary $\Omega = \{\Omega_l\}_{l=1}^{L}$. At each l^{th} layer, we construct a set of part realizations $R^l(s_n) = \{R_i^l(s_n) = \left(G_i^l(s_n), Y_i^l(s_n)\right)\}_{i=1}^{B'_l}$ and an object graph $\mathbb{G}_l = (\mathbb{V}_l, \mathbb{E}_l)$ of s_n, $\forall l = 1, 2, \ldots, L$.

At the first layer $l = 1$, the nodes of the instance graph $G_i^1(s_n)$ of a part realization $R_i^1(s_n)$ represent the Gabor features $f_{na}^i(\mathbf{x}_{na}) \in \hat{F}_n^{te}$ observed in the image $s_n \in S^{te}$ at an image location $\mathbf{x}_{na}$ as described in Section 2.2.

In order to infer the graph instances and compositions of part realizations in the following layers $1 < l < L$, we employ a graph matching algorithm that constructs $G_i^{l+1}(s_n) = \{H(\mathcal{P}^l) : H(\mathcal{P}^l) \subseteq \mathbb{G}_l\}$ which is a set of subgraph isomorphisms $H(\mathcal{P}^l)$ of part graphs $\mathcal{P}^l$ computed in $\mathbb{G}_l$ using an indexing mechanism.

3 Experiments

We examine our proposed approach on three benchmark object shape datasets, which are namely the Amsterdam Library of Object Images (ALOI) [9], the Tools and the Myth [2]. In the experiments, we used $\Theta = 6$ number of different orientations of Gabor features with the same Gabor kernel parameters implemented in [7]. We used subsampling ratio as $\sigma = 0.5$.

3.1 Experiments on Multiple-View Images

ALOI dataset consists of multiple view images of objects belonging to 1000 categories. Each view of an object is captured by rotating the object by 5^o starting from a reference view point labelled as 0^o to 355^o. In the experiments, we used 14 images captured from the viewpoints labelled $\{25^o, 50^o, 75^o, \ldots, 350^o\}$ as test images, and 14 images captured from the viewpoints labelled $\{30^o, 55^o, 80^o, \ldots, 355^o\}$ as training images, for each object.

In the first set of experiments, we analyzed the part shareability and computational complexity of the algorithms across multiple view images of a cup and a duck. For each layer $l = 1, 2, 3, 4, 5$, part realizations and object graphs detected on multiple view cup images and duck images are shown in Table 1 and Table 2, respectively. In the images, each part with a different part realization id is depicted by a different color. For instance, for an image of a cup captured from a viewpoint labelled as 75^o, there are 6 different types of parts with 78 different part realizations at the first layer $l = 1$ (see second column of Table 1). However, we observe 5 different types of part compositions at the fifth layer $l = 5$ of the hierarchy. In the results, each node of an object graph, which is visualized by red points and lines, represents a position of center of a part.

In the analyses of graph structures, we observe that the locality of topological structures of object graphs decreases through the higher layers representing object shapes with higher abstraction. For instance, part realizations of the parts represented with Gabor features at the first layer are connected to each other in a spatial neighbourhood in the results shown at $l = 1$ and $l = 2$ in Table 1 and Table 2. However, the connectivity of part realizations are determined using statistical and descriptive relationships between parts at the higher layers; horizontally oriented part realizations detected at the top and bottom of a cup and a duck are connected to each other, and vertically oriented part realizations detected at the right and left of the cup and duck are connected to each other for $l \geq 3$ in Table 1 and Table 2.

Table 1: Results on multiple view cup images obtained from ALOI Dataset

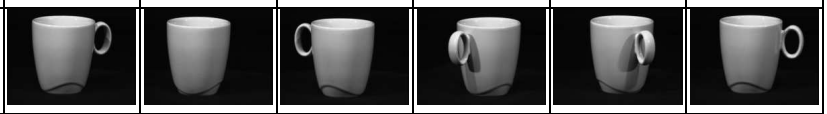
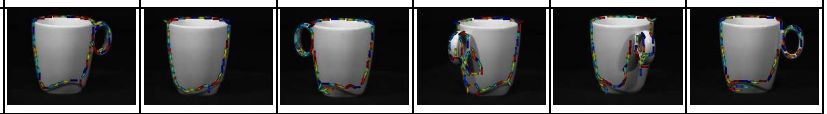
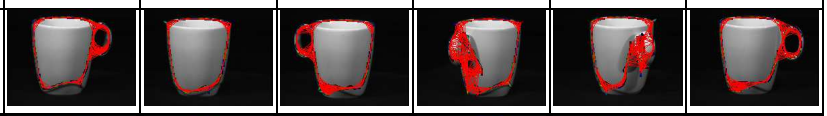
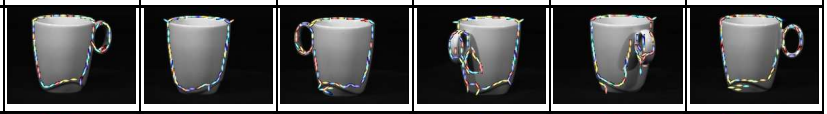
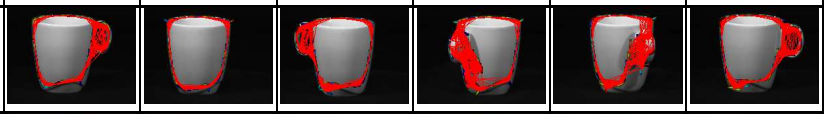
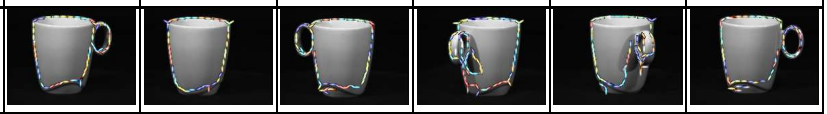
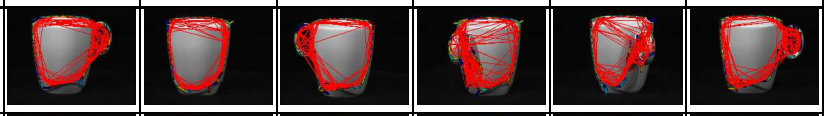
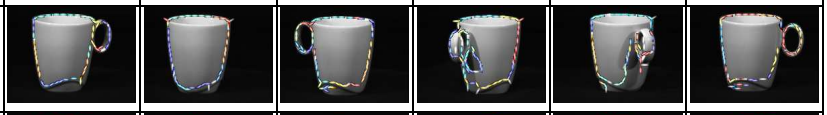
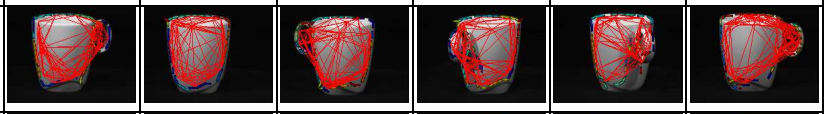
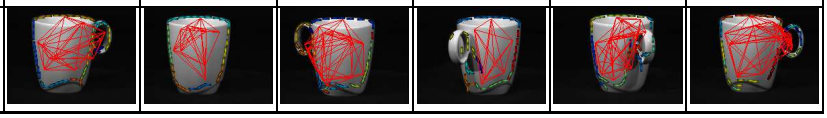
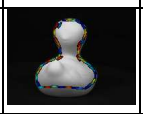
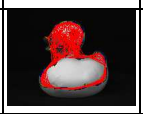
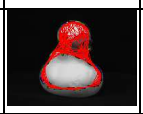
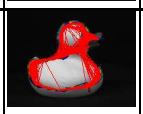
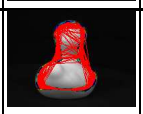
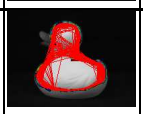
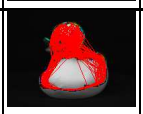
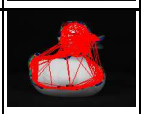
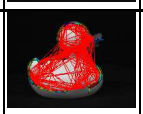
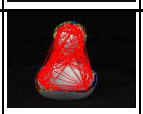
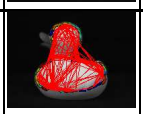
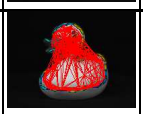
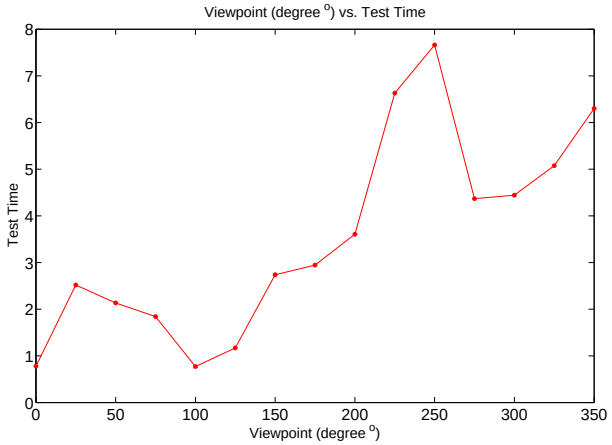
Rotation Degree	25^o	75^o	150^o	225^o	300^o	350^o
Original Image						
Layer $l = 1$						
Object Graph at $l = 1$						
Layer $l = 2$						
Object Graph at $l = 2$						
Layer $l = 3$						
Object Graph at $l = 3$						
Layer $l = 4$						
Object Graph at $l = 4$						
Layer $l = 5$						
Object Graph at $l = 5$						

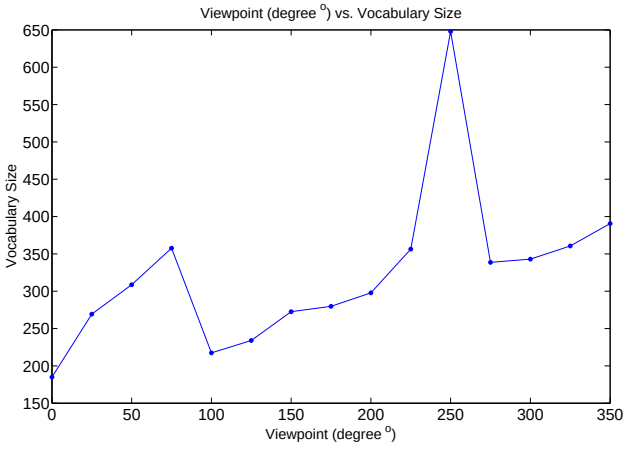
Table 2: Results on multiple view duck images obtained from ALOI Dataset

Rotation Degree	25^o	75^o	150^o	225^o	300^o	350^o
Original Image						
Layer $l = 1$						
Object Graph at $l = 1$						
Layer $l = 2$						
Object Graph at $l = 2$						
Layer $l = 3$						
Object Graph at $l = 3$						
Layer $l = 4$						
Object Graph at $l = 4$						
Layer $l = 5$						
Object Graph at $l = 5$						
Layer $l = 6$						
Object Graph at $l = 6$						

In the second set of experiments, we analyzed the change of inference time in testing phase, and shareability of parts across different views of objects as new images captured at different viewpoints are added to training and test datasets. In the experiments, initially there is only one image captured from a viewpoint labelled 25^o degree and 30^o degree in a test and training dataset, respectively. Then, new images captured from the viewpoints labelled with a value obtained from $\{50^o, 75^o, \ldots, 350^o\}$ and $\{55^o, 80^o, \ldots, 355^o\}$ are sequentially added to test and training datasets, respectively. Analyses for a cup and a duck are given in Fig. 4 and 5.

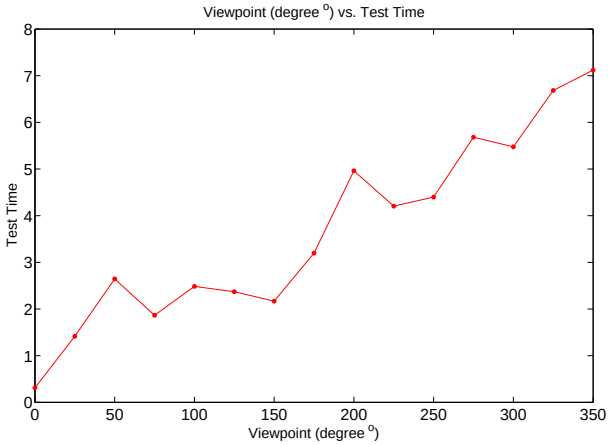


(a) Viewpoint ($degree^0$) vs. Inference time in testing phase.

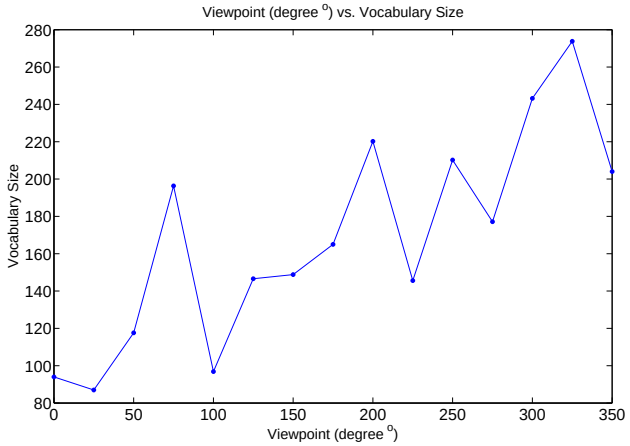


(b) Viewpoint ($degree^0$) vs. Vocabulary size ($|\Omega|$).

Fig. 4: Experimental analyses for a cup in the ALOI dataset.



(a) Viewpoint ($degree^0$) vs. Inference time in testing phase.



(b) Viewpoint ($degree^0$) vs. Vocabulary size ($|\Omega|$).

Fig. 5: Experimental analyses for a duck in the ALOI dataset.

In Fig. 4 and 5, vocabulary size $|\Omega|$ decreases as part shareability increases. This is due to the proposed part selection and composition methods which first employ statistical learning of part distributions in order to learn the statistical relationships between parts. Then, the learned relationships are used to compute *description length* of parts, and an MDL based compression method is employed for construction of compositions of parts. For instance, a value of $|\Omega|$ computed at the viewpoint 75^o decreases when a new image captured at a viewpoint 100^o is used to incrementally learn the vocabulary of the CHOP in Fig. 4.b. The reason is that we observe a *smooth* shape boundary of the cup without a handle part when an image is captured at the viewpoints 75^o and 100^o. Then, the

co-occurrence frequency values of the parts that represent the smooth shape boundary increase and they reside in the same clusters with *less* conditional entropy leading to object graphs with *smaller* description length compared the observations at viewpoints 25^o and 50^o. Therefore, the parts representing the smooth shape boundaries are compressed and encoded according to both their statistical relationships and description length values.

Additionally, we observe that the inference time in testing phase decreases as the vocabulary size decreases and shareability of parts across images captured at different views of objects increases in Fig. 4.a and 5.a. The relationship between inference time and $|\Omega|$ is observed because of the indexing mechanism used in the implementation of the inference algorithm. Note that the proposed part composition method based on a data compression process enables us to use part shareability property to decrease the inference time in testing phase.

3.2 Experiments on Partial Shape Similarity

Employing part shape similarity for learning composition of parts is an important requirement for hierarchical compositional architectures [8]. In this section, we examine this property of the proposed CHOP algorithm in an articulated shape dataset called the Myth dataset [2].

In the Myth dataset, there are three categories, namely Centaur, Horse and Man. There are 5 different images belonging to 5 different objects in each category. Shapes observed in images differ by additional parts, e.g. the shapes of objects belonging to Centaur and Man categories share the upper part of the man body, and the shapes of objects belonging to Centaur and Horse categories share the lower part of the horse body. In the experiments, four samples belonging to each category is used for training and the other three images are used for testing. The results of four experiments are shown in Table 3, 4, 5 for Centaur, Horse and Man category, respectively. The results are shown for the last two layers that are achieved in the construction of object graphs for each shape. In the tables, the right column labeled $l+1$ represents the top layer, and the left column labeled l represent the previous column. For instance, the left column of Centaur-1 shape depicts part realizations and object graphs detected at the layer $l = 7$, and the right column depicts part realizations and object graphs detected at the layer $l+1 = 8$ of the hierarchy in Table 3. Note that top layers of inference trees at which part realizations and object graphs are detected can be different for different shapes and images, since a hierarchical vocabulary and inference trees are dynamically constructed in the CHOP.

In the experiments, we first observe that the depths of inference trees of objects belonging to the same category are closer to each other than that of objects belonging to different categories. For instance, depths of inference trees of 3 Centaur shapes are 8 and that of one Centaur shape is 7. Meanwhile, depths of inference trees of 3 Man shapes are 6 and that of one Man shape is 7.

Moreover, we observe that the shared parts are correctly detected in part realizations and successfully employed in the construction of compositions. For instance, legs of horses which are shared among Centaur and Horse categories

are represented as single compositions in the vocabularies and detected as realizations with unique id at the top layer of the inference trees. However, back parts of horses are depicted with different shapes, therefore these parts are not shared across categories. Consequently, the unshared parts are not detected in the inference trees and not used in the construction of part vocabularies. Similarly, the articulated right arms of man shapes which are shared in 5 shapes belonging to Man and Centaur categories are detected in the inference trees.

Table 3: Results on images belonging to Centaur category obtained from Myth Dataset

Object Name, *Layer ID*	l		$l+1$	
	Part Realizations R^l	Object Graph $\mathbb{G}_l$	Part Realizations R^{l+1}	Object Graph $\mathbb{G}_{l+1}$
Centaur-1, $l = 7$				
Centaur-2, $l = 7$				
Centaur-3, $l = 7$				
Centaur-4, $l = 6$				

Table 4: Results on images belonging to Horse category obtained from Myth Dataset

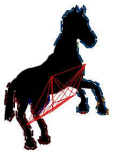
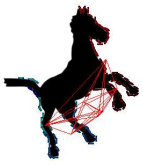
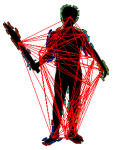
Object Name, *Layer ID*	l		$l+1$	
	Part Realizations R^l	Object Graph $\mathbb{G}_l$	Part Realizations R^{l+1}	Object Graph $\mathbb{G}_{l+1}$
Horse-1, $l = 7$				
Horse-2, $l = 7$				
Horse-3, $l = 6$				
Horse-4, $l = 6$				

Table 5: Results on images belonging to Man category obtained from Myth Dataset

Object Name, *Layer ID*	l		$l+1$	
	Part Realizations R^l	Object Graph $\mathbb{G}_l$	Part Realizations R^{l+1}	Object Graph $\mathbb{G}_{l+1}$
Man-1, $l = 5$				
Man-2, $l = 5$				
Man-3, $l = 5$				
Man-4, $l = 6$				

3.3 Experiments on Articulated Shape Images

In the last set of experiments, we examined the proposed approach using articulated Tools dataset [2]. The dataset consists of 35 shapes belonging to 4 categories. Images belonging to Scissor and Pliers categories are used in the experiments. In each experiment, we selected one object belonging to a category as a training object and the other object in the same category as a test object. An articulation is used to construct different shapes of objects. Experiments on Scissor and Pliers categories are shown in Table 6 and 7, and Table 8 and 9, respectively. For instance, images belonging to Scissor-2 are used for training a vocabulary of a CHOP for detection of parts of shapes in images belonging to Scissor-1 in the experiments given in Table 6, and vice versa in Table 7.

In the results, junctions and closed curves observed at the shape boundaries are detected as part realizations, if they are shared among different articulated images. Moreover, these shape parts are represented as single part compositions at the top layers of inference trees by object graphs. For instance, circular shape handles of scissors and V shaped handles of pliers are represented as compositions with unique id in Table 6 and 7, and Table 8 and 9, respectively.

Table 6: Results on images of Scissor-1 object belonging to Scissor category obtained from Tools Dataset

Object Name, **Articulation ID**, *Layer ID*	l		$l+1$	
	Part Realizations R^l	Object Graph $\mathbb{G}_l$	Part Realizations R^{l+1}	Object Graph $\mathbb{G}_{l+1}$
Scissor-1, Art-1, $l = 6$				
Scissor-1, Art-2, $l = 6$				
Scissor-1, Art-3, $l = 6$				
Scissor-1, Art-4, $l = 5$				
Scissor-1, Art-5, $l = 6$				

Table 7: Results on images of Scissor-2 object belonging to Scissor category obtained from Tools Dataset

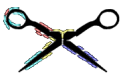
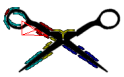
Object Name, **Articulation ID**, *Layer ID*	l		$l+1$	
	Part Realizations R^l	Object Graph $\mathbb{G}_l$	Part Realizations R^{l+1}	Object Graph $\mathbb{G}_{l+1}$
Scissor-2, Art-1, $l = 5$				
Scissor-2, Art-2, $l = 5$				
Scissor-2, Art-3, $l = 5$				
Scissor-2, Art-4, $l = 5$				
Scissor-2, Art-5, $l = 5$				

Table 8: Results on images of Pliers-1 object belonging to Pliers category obtained from Tools Dataset

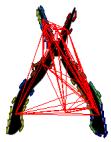
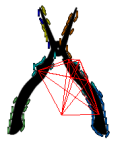
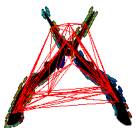
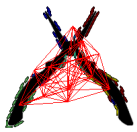
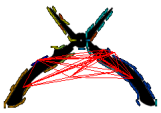
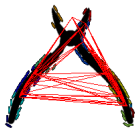
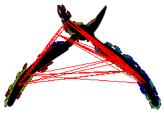
Object Name, **Articulation ID** *Layer ID*	l		$l+1$	
	Part Realizations R^l	Object Graph $\mathbb{G}_l$	Part Realizations R^{l+1}	Object Graph $\mathbb{G}_{l+1}$
Pliers-1, Art-1, $l = 5$				
Pliers-1, Art-2, $l = 4$				
Pliers-1, Art-3, $l = 5$				
Pliers-1, Art-4, $l = 5$				
Pliers-1, Art-5, $l = 5$				

Table 9: Results on images of Pliers-2 object belonging to Pliers category obtained from Tools Dataset

Object Name, **Articulation ID** *Layer ID*	l		$l+1$	
	Part Realizations R^l	Object Graph $\mathbb{G}_l$	Part Realizations R^{l+1}	Object Graph $\mathbb{G}_{l+1}$
Pliers-2, Art-1, $l = 5$				
Pliers-2, Art-2, $l = 5$				
Pliers-2, Art-3, $l = 5$				
Pliers-2, Art-4, $l = 5$				
Pliers-2, Art-5, $l = 5$				

4 Conclusion

We have proposed a graph theoretical approach for object shape representation in a hierarchical compositional architecture called Compositional Hierarchy of Parts (CHOP). In the proposed approach, vocabulary learning is performed using a hybrid generative-descriptive model. Two information theoretic algorithms are used for learning a vocabulary of compositional parts. First, statistical relationships between parts are learned using a *Minimum Conditional Entropy Clustering* algorithm. Then, part selection problem is defined as a Subgraph Isomorphism problem, and solved using an MDL principle. Part compositions are inferred considering both learned statistical relationships between parts and their description length at each layer $1 < l \leq L$ in an L-layer CHOP.

The proposed approach and algorithms are examined using a multiple view image dataset and two articulated image datasets. In the experiments performed using multiple view image datasets, we examined part shareability property and inference time complexity of CHOP across different images of an object captured at different viewpoints. The results show that CHOP can recognize and use part shareability property in the construction of vocabularies and inference trees. For instance, if the parts of shapes encoded in a learned vocabulary and a new given shape, which will be used for incremental learning of vocabulary, are shareable, then the shareable parts can be used to improve the statistical relationships between learned parts, and minimize description length of parts and compositions in the CHOP.

Two types of experiments are performed on articulated images. In the first group, we used the Myth dataset consisting of shapes each of which share some parts with the other shapes in the dataset. The analyses show that *most frequently* shared parts are successfully used in the construction of vocabularies and detected on images. For instance, legs of horses which are shared among Centaur and Horse categories are detected as realizations of single compositions at the top layer of the inference trees. However, back parts of horses are depicted with different shapes, therefore these parts are not shared across categories. In the second group, we used the Tools dataset which contains images that differ by an articulation. The results show that junctions and closed curves observed at the shape boundaries can be detected as part realizations if they are shared among different articulated images.

In the future work, we will employ discriminative learning for pose estimation and categorization of shapes. In addition, online and incremental learning will be used considering the results obtained from the analyses on part shareability performed in this work.

Acknowledgement

This work was supported by the European commission project PaCMan EU FP7-ICT, 600918.

References

1. Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug 2013.
2. A. M. Bronstein, M. M. Bronstein, A. M. Bruckstein, and R. Kimmel, "Analysis of two-dimensional non-rigid shapes," *Int. J. Comput. Vision*, vol. 78, no. 1, pp. 67–88, Jun 2008.
3. D. J. Cook and L. B. Holder, *Mining Graph Data*. John Wiley & Sons, 2006.
4. R. Davies, C. Twining, T. Cootes, J. Waterton, and C. Taylor, "A minimum description length approach to statistical shape modeling," *IEEE Trans. Med. Imag.*, vol. 21, no. 5, pp. 525–537, May 2002.
5. P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep 2010.
6. P. Felzenszwalb and J. Schwartz, "Hierarchical matching of deformable shapes," in *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.
7. S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, ser. CVPR '07, June 2007, pp. 1–8.
8. S. Fidler, M. Boben, and A. Leonardis, "Learning hierarchical compositional representations of object structure," in *Object categorization computer and human perspectives*, S. J. Dickinson, A. Leonardis, B. Schiele, and M. J. Tarr, Eds. Cambridge, UK: Cambridge University Press, 2009, pp. 196–215.
9. J.-M. Geusebroek, G. Burghouts, and A. Smeulders, "The amsterdam library of object images," *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 103–112, 2005.
10. I. Kokkinos and A. Yuille, "Inference and learning with hierarchical shape models," *Int J Comput Vis*, vol. 93, no. 2, pp. 201–225, 2011.
11. A. Levinshtein, C. Sminchisescu, and S. Dickinson, "Learning hierarchical shape models from examples," in *Proceedings of the 5th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*, ser. EMMCVPR'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 251–267.
12. H. Li, K. Zhang, and T. Jiang, "Minimum entropy clustering and applications to gene expression analysis," in *Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, ser. CSB '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 142–151.
13. B. Ommer and J. Buhmann, "Learning the compositional nature of visual object categories for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 3, pp. 501–516, Mar 2010.
14. R. Salakhutdinov, J. Tenenbaum, and A. Torralba, "Learning with hierarchical-deep models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1958–1971, Aug 2013.
15. A. Torsello and E. Hancock, "Learning shape-classes using a mixture of tree-unions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 954–967, June 2006.
16. Z. Xu, H. Chen, S.-C. Zhu, and J. Luo, "A hierarchical compositional model for face representation and sketching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 955–969, June 2008.

17. L. Zhu, Y. Chen, and A. Yuille, "Learning a hierarchical deformable template for rapid deformable object parsing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1029–1043, June 2010.
18. L. L. Zhu, Y. Chen, and A. Yuille, "Recursive compositional models for vision: Description and review of recent work," *J. Math. Imaging Vis.*, vol. 41, no. 1-2, pp. 122–146, Sep 2011.

Object Categorization from Range Images using a Hierarchical Compositional Representation

Vladislav Kramarev
School of Computer Science
University of Birmingham
vvk201@cs.bham.ac.uk

Sebastian Zurek
School of Computer Science
University of Birmingham
s.j.zurek@cs.bham.ac.uk

Jeremy L. Wyatt
School of Computer Science
University of Birmingham
j.l.wyatt@cs.bham.ac.uk

Aleš Leonardis
School of Computer Science
University of Birmingham
a.leonardis@cs.bham.ac.uk

***Abstract*—This paper proposes a novel hierarchical compositional representation of 3D shape that can accommodate a large number of object categories and enables efficient learning and inference. The hierarchy starts with simple pre-defined parts on the first layer, after which subsequent layers are learned recursively by taking the most statistically significant compositions of parts from the previous layer. Our representation is able to scale because of its very economical use of memory and because subparts of the representation are shared. We apply our representation to 3D multi-class object categorization. Object categories are represented by histograms of compositional parts, which are then used as inputs to an SVM classifier. We present results for two datasets, Aim@Shape [1] and the Washington RGB-D Object Dataset [2], and demonstrate the competitive performance of our method.**

***Keywords*—3D object representation, 3D object categorization, compositional hierarchy, classification.**

I. Introduction

Reliable object recognition and categorization has been one of the central topics addressed by the computer vision community over decades. The methods based on visual words are widely used to solve 3D object categorization and shape retrieval problems. Some authors, for example Toldo et al. [3] and Fehr et al. [4], use a Bag-of-Words (BoW) strategy where an object is represented by a set of local features. Others, e.g. Madry et al. [5], also introduce data structures describing spatial relations of local features.

Compositional hierarchies have recently become a popular topic in computer vision. Principles of *hierarchical compositionality* allow one to develop generalizable category representation and recognition frameworks, where new categories can be added efficiently to the system. However, most of the recent advances in this area have been focused on hierarchies of 2D features [6][7][8][9][10][11]. Very little work has been done so far to address the formidable problem of true 3D categorization using compositional hierarchical approaches.

In this paper we shed some light on this problem and propose a hierarchical compositional representation of 3D shapes that is a recursive compositional vocabulary of surface parts represented by a directed graph [7] (see Figure 1).

The first layer L_1 of the hierarchy contains several pre-defined parts. All the parts from the above layers are learned and represent the most statistically significant compositions of several simpler shape parts from bottom layers.

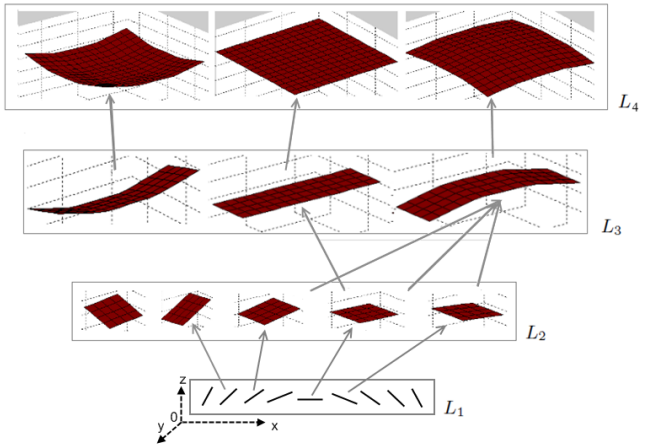


Fig. 1. 3D compositional hierarchy of parts.

In order to examine the learned layers of the compositional hierarchy, we introduce a new 3D object categorization method which is based on histograms of compositional parts. Each object category is represented by histograms reflecting the spatial distribution of the compositional parts that describe the object's surface. We employ an SVM classifer with χ^2 kernels for categorization. We tested our method on the Aim@Shape dataset [1] containing 20 object categories and achieved 95.6% success rate for categorization. We also obtained promising results for the larger Washington RGB-D Object Dataset [2].

The rest of the paper is organized as follows. In Section II we discuss related work. In Section III we give a detailed description of our method. Section IV describes our experiments and results. Section V concludes the paper.

II. Related work

The proposed method is related to the works of Fidler et al. [6][7][8]. They have introduced a framework for learning a hierarchical compositional shape vocabulary for multi-class object representation. Each part in the hierarchy is composed of *less complex* parts according to statistical properties of their spatial configurations. At each layer, parts are recursively combined into more complex compositions, each exerting a high degree of shape variability. At the top layer of the hierarchical vocabulary, the compositions are sufficiently complex

to represent the shape of a whole object. The main difference between our proposed work and Fidler et al. is that they provided a mechanism to learn a 2D shape vocabulary (contour parts), while our proposed method represents 3D shapes of objects in a compositional hierarchy.

Savarese et al. [12] introduced a hierarchical framework for 3D object categorization and recognition. They extracted local features from the images and grouped them into relatively large discriminative regions (called parts) that are pulled together to form a 3D category model. Whilst our compositional hierarchy models 3D shape independently of 2D image context, Savarese et al. built a hierarchy grouping both 2D local features and 3D shape features This precludes their approach for applications where only 3D data is provided (e.g. Kinect data, or haptic data in robotics applications).

Pratikakis et al. [13] proposed a 3D compositional model in which point clouds are decomposed into sections that are represented by a predefined set of primitives, e.g. cone, torus, sphere or cylinder. Their method has a limitation in that it deals with the simplest shapes (mainly hand-made objects) and hence is not suitable for general multi-class category detection.

Detry et al. [14] proposed a hierarchical object representation framework that encodes probabilistic spatial relations between 3D features using Markov networks. Features extracted at the base layer of the hierarchy are bound to local 3D descriptors. Higher-level features recursively encode probabilistic spatial configurations of the features obtained from previous layers. However, their approach does not involve statistical learning of a single 3D shape vocabulary that is shared by objects of different categories.

Recently Fox and colleagues have published a series of papers in which they introduced several algorithms for object classification (at both category and instance level) and evaluated these on their RGB-D image dataset. In [2], after partitioning the depth image within a 3D bounding box, they computed spin image [15] histograms that were used to form *efficient match kernel* (EMK) features. After dimensionality reduction with PCA, these features (around 2700) were used to train a classifier, such as a gaussian-kernel SVM. In subsequent work Lai et al. [16] developed a new classifier, based on the *instance distance learning* (IDL) technique and data sparsification, that was able to improve categorization performance. By using kernel descriptors and *hierarchical matching pursuit* to build feature hierarchies, further gains in categorization accuracy were achieved [17] [18]. For comparison with our method, we show their results for object category recogntion from range images in Table II.

III. Compositional Hierarchy of 3D Parts

In this section we describe our compositional hierarchical representation of 3D object shape, how the representation is learned and how to perform inference using it.

A. Representation

We define our coordinate system such that the x and y axes span the image, and the z-axis encodes depth information. We define a hierarchy of layers, where L_n denotes the n-th layer of the hierarchy. The first layer L_1 of the hierarchy contains several pre-defined, rather than learned, features or parts. First layer parts encode quantized differences of depth (relative depth) between pixels at a fixed distance from each other in the x-axis direction. Figure 2 shows one way in which these parts can be defined. In this case we quantize all possible values of relative depth into nine bins. However, the number of first layer parts can be chosen differently depending on the type of input data and required precision of the representation.

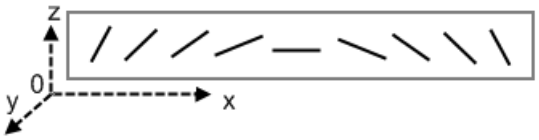


Fig. 2. Parts of the first layer.

Figure 3c shows the range image of the mug and demonstrates how the range data can be encoded in terms of the pre-defined first layer parts. In Figure 3d locations of parts depicted in Figure 3a are represented using color coding given in Figure 3b.

Fig. 3. (a) Pre-defined first layer parts. (b): Color coding of the first layer parts. (c): Range image of the mug. (d): Encoding of the mug with first layer parts.

In general, the higher layers L_n, $\forall n > 1$, are learned using joint statistical properties of parts from the layer below. Each part P_i^n in L_n is a composition of subparts, that is a list of subparts and a description of the spatial relations between these constituent subparts. We say that a composition P_i^n consists of a central part $P_{central}^{n-1}$ and other subparts that reside at some positions relative to $P_{central}^{n-1}$:

$$P_i^n \equiv (P_{central}^{n-1}, \{P_j^{n-1}, \mu_j, \Sigma_j\}_j) \quad (1)$$

where $\mu_j = (x_j, y_j, z_j)$ is the mean relative position of the subpart P_j^{n-1}, and Σ_j is the covariance matrix expressing the variability of possible relative positions. In this paper, we specialize this scheme by assuming that all compositions consist of three subparts.

Second layer parts can be regarded as very small surface patches, that are constructed out of three L_1 parts. Figure 4 sketches the construction of second layer parts (in the general scheme), and Figure 5 illustrates several examples of learned parts.

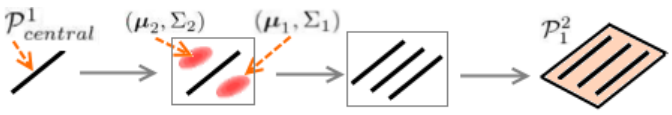


Fig. 4. Construction of second layer parts.

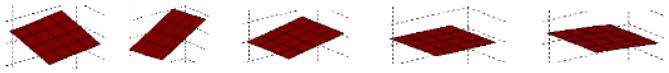


Fig. 5. Examples of second layer parts.

Third layer parts are assembled from triplets of L_2 parts, which are adjacent along the x-direction. Similarly fourth layer parts consist of three adjacent L_3 parts aligned vertically (i.e. along the y-direction).

In this and future work we intend to demonstrate that the representation proposed above has the following desirable properties:

Efficient use of memory: In current state-of-the-art 3D object categorization, objects are mainly represented by salient surface patches or discriminative local features such as spin images [15], or by statistical moments. Very large numbers of patches and local features must be collected and stored in order to achieve the best results for multi-class categorization. In compositional hierarchical approaches, parts are shared throughout the hierarchy. More complex parts are described in terms of simpler parts from the previous layers, therefore a simpler part at one layer can be used to describe many parts in higher layers. This re-usability results in a very compact representation of the vocabulary.

Unsupervised learning: Compositional parts in the hierarchy are learned in an unsupervised manner. This learned "vocabulary of parts" captures and compactly represents the most statistically relevant regularities in the dataset.

Fast, incremental learning: The proposed method enables new object categories to be learned efficiently, i.e. with less computational complexity than batch schemes. Moreover its efficiency increases with the amount of data already learned by the system. In fact, new objects or object categories can be added to the representation by simply pulling together a small number of appropriate parts.

B. Learning a vocabulary of parts

The goal of the learning procedure is to construct compositions of parts that encode the most statistically significant spatial relations between parts of the layer below. The collection of compositions from all layers in a trained compositional hierarchy is termed a vocabulary. In general, each composition has to be flexible in that it should tolerate some variability in the relative spatial position of elements.

The learning process for each layer L_n, $\forall n > 1$, can be summarized in four steps:

1) Perform *local inhibition* in the neighborhood of each part.
2) Construct statistical maps that characterize the 3D spatial relations between parts of the previous layer.
3) Produce a list of *candidate parts* by constructing compositions based on the statistical maps.
4) Optimize the list of candidate parts to form the vocabulary, i.e. select a subset of parts that satisfies some optimality criterion

We now describe each step in more detail.

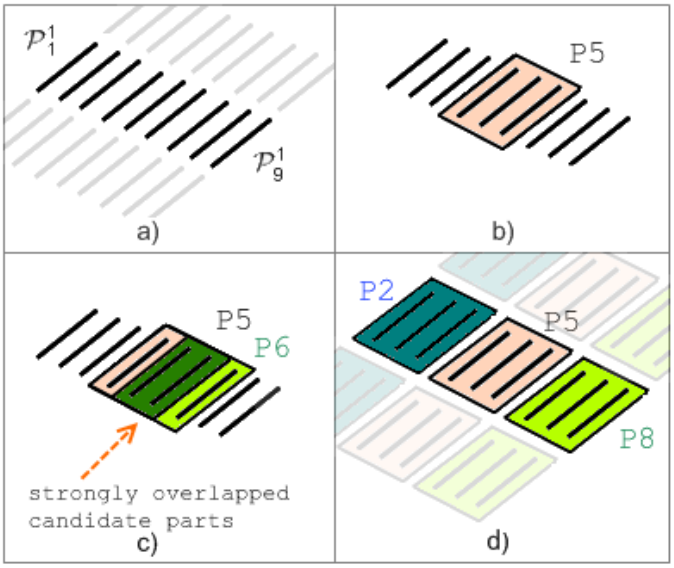


Fig. 6. Implementation of local inhibition: a) Detected parts of layer L_1 lying on a surface; b) Derived part P_1^2 of the L_2 layer; c) Other L_2 parts that have intersection with P_1^2 are to be removed (e.g. part P_4^2); d) Surface patch covered by L_2 parts after performing local inhibition.

The important first step is local inhibition which helps to avoid unnecessary redundancy in coding. Assume that we are given a range image that is encoded in terms of parts $\{P_j^n\}$ at layer L_n. For each part P_k^n, we remove the parts that reside in a (small) neighborhood of P_k^n and have a large intersection with P_k^n in terms of L_{n-1} parts.

This step can be considered as the removal of those local surface features that are already partially encoded by P_k^n. The procedure is illustrated in Figure 6 for L_2 parts.

Next, we construct statistical maps that describe relative positions of parts in 3D space. The maps for layer L_n are functions f:

$$f(P_i^{n-1}, P_j^{n-1}, x, y, z) \rightarrow [0, 1] \qquad (2)$$

that are defined for each pair of elements P_i^{n-1} and P_j^{n-1} in layer L_{n-1}, and a 3D offset $(x, y, z) \in \mathbb{R}^3$. The maps encode the probability of observing a part P_j^{n-1} displaced by (x, y, z) relative to a central part P_i^{n-1}. A natural way to visualize the collected co-occurrence statistics is to project the 5-dimensional function f into 3 dimensions by fixing the first and second parameters.

After the co-occurrence statistics of parts are computed, we detect peaks in the spatial maps, and fit the data in surrounding regions by a Gaussian distribution with mean μ_j and covariance matrix Σ_j. Figure 7 shows an example of such fitting of the statistical map depicting co-occurrences of the second layer parts P_{41}^2 and P_{42}^2.

Parts P_i^n of the layer L_n are constructed from the previous layer L_{n-1} using μ_j and Σ_j as shown in (1). This procedure is implemented in two steps. First, we construct pairs, i.e. elements comprising two parts from the previous layer. Next, we group them into triples encoding spatial relations of three parts from the previous layer. Triples become *candidate parts*

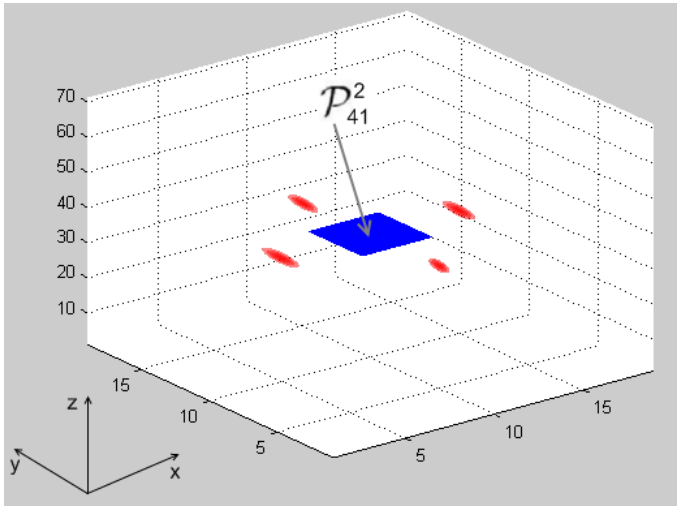


Fig. 7. Statistical map depicting co-occurrences of parts P^2_{41} and P^2_{42}. The size of the local neighborhood was chosen to be $17 \times 17 \times 70$.

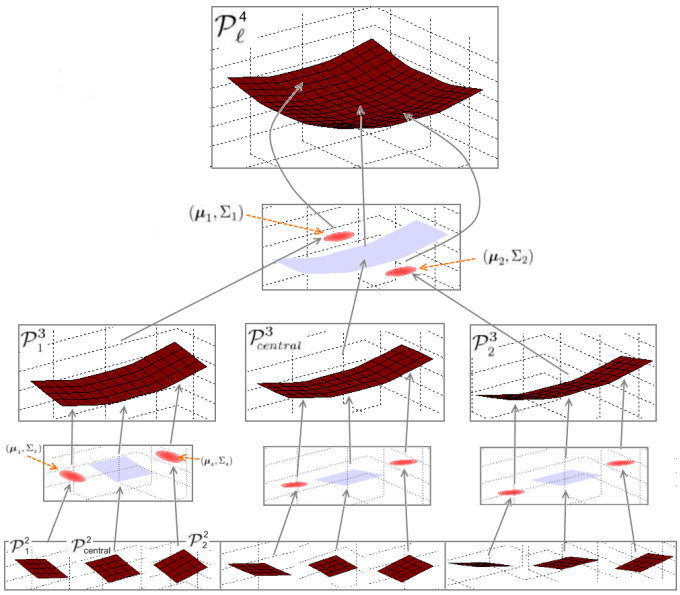


Fig. 8. General learning scheme for layers L_3 and L_4 in the compositional hierarchy.

that will reside in layer L_n. Figure 8 depicts how triples are formed for the third and fourth layers.

Our experiments have shown that, for the first layers of the hierarchy, steps 2 and 3 in the above algorithm can be approximated. It is sufficient to consider only the z component that has a predefined spatial relationship in the x and y directions defined by the object size. Hence, for the experiments in this paper, we assumed that parts (in a given layer) had a predefined spatial relationship in the x and y directions, and we collected only a quantized z component of the 3D offset. This simplification did not significantly affect the part selection process and therefore the categorization accuracy. However, it yielded a significant improvement in terms of processing time. We note that for learning layers beyond L_4 this simplification may not be suitable, as more complex parts may have more complex spatial configurations.

Typically the set of candidate parts $S = \{P^n_i : i = 1..N\}$ for the given layer L_n is rather large, and contains many parts that represent very similar surface types. In order to maintain a manageable number of parts in the vocabulary and to facilitate generalization, we specify a procedure that selects a somewhat smaller subset $S' \subseteq S$. This selection is performed by approximately solving the following optimization problem. The cost function E which is minimized measures the reconstruction error, i.e. how well the set of candidate parts can be represented by the vocabulary. We also include a term that penalizes vocabularies with more parts, so that the cost E takes the following form:

$$E(S') = \min_{S' \in S} \sum_{i=1}^{N} d(P_i, P'(P_i))\, \nu_i \; + \; \alpha\, |S'| \qquad (3)$$

where ν_i is the frequency of occurrence of the i-th candidate part P^n_i, $d(\cdot,\cdot)$ is a distance function that quantifies the similarity between two parts (from the same layer), and $P'(P_i)$ is the part in S' that is closest to P_i. Also $\alpha \in \mathbb{R}^+$ is a meta-parameter that regulates the trade-off between precision of the representation and number of selected parts. In addition, we have explored adding further penalty terms to influence part selection according to the geometric properties of parts.

C. Inference

This section describes the inference process that generates features from a range image, using a given vocabulary. These features can then be used for object category (or instance) recognition.

Our method performs part detection layer by layer, starting from the first layer. Assume we are given a range image of an object (or scene), where each pixel value encodes depth. The goal is to represent the object in terms of parts from the compositional hierarchical shape vocabulary.

The first stage is to represent the object in terms of first layer parts. We convolve an oriented Gaussian-derivative filter (aligned along the x-axis) with the range image. The variance parameter σ associated with the filter depends on the noise level of the images and was chosen from within the range [0.5,2.0].

The next stage is to quantize the filter response at each pixel by assignment to the bin that corresponds to the closest first layer part. A reconstruction error E_i is computed as a distance to the closest bin center divided by the size of the bin. This procedure gives us a set of *potential parts* $S_{pot} = \{P_1, P_2, ..., P_m\}$, that can be detected at certain locations with corresponding reconstruction errors $E_1, E_2, ..., E_m$, where m is the number of detected potential parts.

However, such a strategy leads to a redundant representation, as all the detected potential parts are significantly overlapped with each other. To proceed, we specify several criteria that our inference process should jointly optimize:

1) Maximize the surface coverage. Ideally the entire object surface should be covered by parts from the vocabulary.
2) Minimize the overlaps between detected parts.

3) Minimize the reconstruction error.

To fulfil all the above requirements we have to select a subset S_{sel} of potential parts S_{pot}. Following e.g. Leonardis et al. [19] we define an energy function that incorporates all three criteria, and then solve the associated optimization problem.

When part detection for the first layer is completed, we do inference at subsequent layers L_n, $\forall n > 1$, performing essentially the same procedure. Suppose we have a range image represented in terms of parts at layer L_{n-1}. Then the inference algorithm for parts at layer L_n can be described as follows:

1) Consider a local neighborhood around each part $\{P_i^{n-1}\}$. For this neighborhood, the part $\{P_i^{n-1}\}$ is referred to as a *central part*.
2) Extract parts located in this neighborhood and their relative positions with respect to the central part ($\{P_i^{n-1}\}$). The central part together with other neighboring parts can be represented as a *potential compositional part* $\{P_{pot}^n\}$ of layer L_n, as described in equation (1).
3) This potential compositional part is matched against vocabulary elements of layer L_n, and if found yields a detection. The matching process can be implemented in a very efficient manner.
4) This procedure leads to a redundant representation, since we attempt to detect a layer L_n part in all positions of detected L_{n-1} parts. Since parts in higher layers are always larger, the potential parts will overlap.
5) We eliminate parts to minimize the reconstruction error, maximize coverage and minimize overlap using an optimization function similar to that described for the first layer.

IV. Experiments

A. Method

In order to evaluate the compositional hierarchical representation we constructed a classifier to perform category-level object recognition from range images. Given a dataset of these images we learn a vocabulary, layer by layer up to L_4. Then we perform multi-class object categorization using histograms of compositional parts, obtained from a training subset of the dataset. Each range image is partitioned into 4 (2×2) and 9 sectors (3×3 – see Figure 9) which together with the original image comprise 14 subimages from which histograms of parts are computed. The histograms are stacked to form a large descriptor which is used as the input vector for each image to a χ^2 kernel SVM classifier.

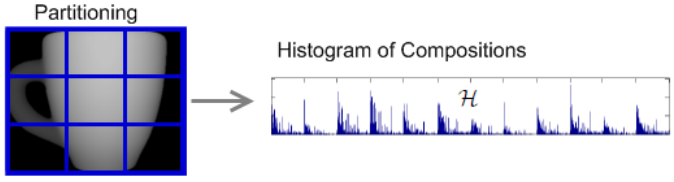


Fig. 9. Partitioning of the object to build a histogram of compositional parts.

We applied this evaluation method to two benchmark datasets: Aim@Shape [1], and the Washington RGB-D Object Dataset [2].

B. Evaluation on Aim@Shape Dataset

From the Aim@Shape dataset we rendered a set of range images presenting all the 3D models at different scales and under different viewing angles. Since the first layers of the hierarchy contain very generic parts which are shared by many categories, only 50-100 of randomly selected range images were required to learn the vocabulary of the second layer, and 200-300 images to learn the third layer.

To be able to compare our approach with other methods we used leave-one-out cross-validation to measure performance. In this experiment we used eight viewing angles and three scales per model to train the SVM classifer.

Figure 10 shows how using features from more layers improves performance.

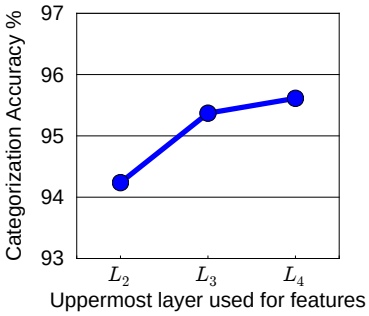


Fig. 10. For the Aim@Shape dataset, the categorization accuracy improves as more layers are used to provide features to the classifier.

In table I we see that the categorization accuracy of our method is comparable with the state-of-the-art, when using features obtained from all layers up to L_4 in the vocabulary.

TABLE I. Results for Aim@Shape Dataset

Method	Accuracy %
Toldo et al. [3]	87.3
Salti et al. [20] using 1-NN for codebooks	79
Salti et al. [20] using 2-NN for codebooks	100
This work (up to L_4)	95.6

C. Evaluation on Washington RGB-D Object Dataset

For the Washington RGB-D Object Dataset we learned a vocabulary up to L_4 from around 2,000 images selected randomly from the whole dataset of about 250,000 images. A remarkable fact is that our shape vocabulary was stored in less than 50kB of memory, demonstrating the memory efficiency of our approach.

To train the SVM classifier, we used only 10% of the available training data. As in [2] we estimated performance with leave-one-out cross-validation.

Table II shows that our method improves upon the earlier work of Lai et al. [2][16] and compares favourably with the accuracies of individual depth kernel descriptors (detailed in [17]).

TABLE II. RESULTS FOR RGB-D OBJECT DATASET (USING RANGE IMAGES ONLY)

Method	Accuracy %
Spin Images & 3D Bounding Boxes [2]	64.7
Sparse Distance Learning [16]	70.2
RGB-D Kernel Descriptors [17]	80.3
Hierarchical Matching Pursuit [18]	81.2
This work (up to L_2)	72.7
This work (up to L_3)	73.8

V. CONCLUSION AND FUTURE WORK

We have presented a 3D learning and recognition framework built on the principle of hierarchical compositionality. The framework accommodates a large number of object categories, and since parts are shared, the size of the representation grows logarithmically with the number of learned object categories. The framework provides mechanisms for *transfer of knowledge* that enables its use in a variety of computer vision and robotics applications, such as object grasping and manipulation.

Thus far we have examined learning for the first four layers of the hierarchy and applied our method to multi-class object categorization with promising results. In future we plan to learn further layers of the compositional hierarchy and to test our method on other 3D categorization and shape retrieval datasets. In particular we intend to investigate how the size of representation changes with the number of object categories. Given the very small memory footprint, the representation may be particularly suited for mobile phone applications.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of EU-FP7-IST grant 600918 (PaCMan). The authors would also like to thank Mete Özay for helpful discussions.

REFERENCES

[1] R. C. Veltkamp and F. B. ter Haar, "SHREC2007: 3D Shape Retrieval Contest," Utrecht University, Tech. Rep. UU-CS-2007-015, 2007.

[2] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *IEEE International Conference on Robotics and Automation, ICRA*, 2011, pp. 1817–1824.

[3] R. Toldo, U. Castellani, and A. Fusiello, "A bag of words approach for 3d object categorization," in *Computer Vision/Computer Graphics CollaborationTechniques*. Springer, 2009, pp. 116–127.

[4] J. . Fehr, A. Streicher, and H. Burkhardt, "A bag of features approach for 3d shape retrieval," in *Advances in Visual Computing*. Springer, 2009, pp. 34–43.

[5] M. Madry, C. H. Ek, R. Detry, K. Hang, and D. Kragic, "Improving generalization for 3d object categorization with global structure histograms," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012*. IEEE, 2012, pp. 1379–1386.

[6] S. Fidler and A. Leonardis, "Towards scalable representations of object categories: Learning a hierarchy of parts," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[7] S. Fidler, M. Boben, and A. Leonardis, "Learning hierarchical compositional representations of object structure," in *Object Categorization: Computer and Human Vision Perspectives*, S. Dickinson, A. Leonardis, B. Schiele, and M. Tarr, Eds. Cambridge University Press, 2009. [Online]. Available: vicos.fri.uni-lj.si/data/alesl/chapterLeonardis.pdf

[8] ——, "Optimization framework for learning a hierarchical shape vocabulary for object class detection." in *BMVC*, 2009, pp. 1–12.

[9] S. C. Zhu and D. Mumford, *A stochastic grammar of images*. Now Publishers Inc, 2007, vol. 2, no. 4.

[10] B. Ommer and J. M. Buhmann, "Learning the compositional nature of visual objects," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07.* IEEE, 2007, pp. 1–8.

[11] L. L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille, "Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 759–773.

[12] S. Savarese and L. Fei-Fei, "3d generic object categorization, localization and pose estimation," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[13] I. Pratikakis, M. Spagnuolo, T. Theoharis, and R. Veltkamp, "Learning the compositional structure of man-made objects for 3d shape retrieval," in *Eurographics Workshop on 3D Object Retrieval (2010)*, 2010.

[14] R. Detry, N. Pugeault, and J. Piater, "A probabilistic framework for 3d visual object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1790–1803, 2009.

[15] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 5, pp. 433–449, 1999.

[16] K. Lai, L. Bo, X. Ren, and D. Fox, "Sparse distance learning for object recognition combining RGB and depth information," in *IEEE International Conference on Robotics and Automation, ICRA*, 2011, pp. 4007–4013.

[17] L. Bo, X. Ren, and D. Fox, "Depth kernel descriptors for object recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2011, pp. 821–826.

[18] ——, "Unsupervised feature learning for RGB-D based object recognition," in *Experimental Robotics - The 13th International Symposium on Experimental Robotics, ISER*, 2012, pp. 387–402.

[19] A. Leonardis, H. Bischof, and J. Maver, "Multiple eigenspaces," *Pattern Recognition*, vol. 35, no. 11, pp. 2613–2627, 2002.

[20] S. Salti, F. Tombari, and L. D. Stefano, "On the use of implicit shape models for recognition of object categories in 3d data," in *Computer Vision–ACCV 2010*. Springer, 2011, pp. 653–666.

Semi-supervised Segmentation Fusion of Multi-spectral and Aerial Images

Mete Ozay*
*School of Computer Science
The University of Birmingham, Edgbaston, Birmingham, B15 2TT, United Kingdom.
Email: m.ozay@cs.bham.ac.uk

***Abstract*—A Semi-supervised Segmentation Fusion algorithm is proposed using consensus and distributed learning. The aim of Unsupervised Segmentation Fusion (USF) is to achieve a consensus among different segmentation outputs obtained from different segmentation algorithms by computing an approximate solution to the NP problem with less computational complexity. Semi-supervision is incorporated in USF using a new algorithm called Semi-supervised Segmentation Fusion (SSSF). In SSSF, side information about the co-occurrence of pixels in the same or different segments is formulated as the constraints of a convex optimization problem. The results of the experiments employed on artificial and real-world benchmark multi-spectral and aerial images show that the proposed algorithms perform better than the individual state-of-the art segmentation algorithms.**

I. Introduction

Image segmentation is one of the most important, yet unsolved problems in computer vision and image processing. Various segmentation algorithms studied in the literature have been applied to segment the objects in images [9], [23], [12]. However, there are two main challenges of their employment.

The first challenge is to *extract* a robust structure, e.g. shape, of the segments by analyzing the outputs of segmentation algorithms when a target segmentation is not available with a training dataset. This challenge has been studied as a *segmentation mining* problem and analyzed as a consensus segmentation problem [10], [17] using an Unsupervised Segmentation Fusion approach by Ozay et al. [15].

The second challenge is the selection of an *appropriate* algorithm with its parameters that provides an *optimal* segmentation which is closer to a *target* segmentation if a target segmentation is available with a training dataset. For this purpose, some of the segments in the segmentation set are expected to represent acquired target objects in the Unsupervised Segmentation Fusion algorithms [15], [10], [17]. In order to relax this assumption, first the error and distance functions of the algorithm should be refined to include these requirements. Therefore, prior information on the statistical properties of the datasets need to be incorporated using supervision. Then, side information about a target segmentation output should be used in the unsupervised segmentation fusion algorithm, which leads to a semi-supervised algorithm. In this work, this challenge has been analyzed by Semi-supervised Segmentation Fusion which incorporates prior and side information obtained from training datasets and expert knowledge to the USF algorithm [15].

Consensus segmentation problem is re-formalized as a semi-supervised segmentation fusion problem and studied using decision fusion approaches [8] with semi-supervised learning [6]. For this purpose, an algorithm called Semi-supervised Segmentation Fusion (SSSF) is introduced for fusing the segmentation outputs (decisions) of base-layer segmentation algorithms by incorporating the prior information about the data statistics and side-information about the content into the USF algorithm [15]. In the SSSF, this is accomplished by extracting the available side information about the targets, such as defining the memberships of pixels for the segments which represent a specific target in images. For this purpose, the side information about the pixel-wise relationships is reformulated and incorporated with a set of constraints in the segmentation fusion problem. In addition, a new distance function is defined for the Semi-supervised Segmentation Fusion by assigning weights to each segmentation.

In order to compute the *optimal* weights, the median partition (segmentation) problem is converted into a convex optimization problem. The side information which represents the pixel-wise segmentation membership relations defined by must-link and cannot-link constraints are incorporated in an optimization problem and in the structure of distance functions. Moreover, sparsity of the weights are used in the optimization problem for segmentation (decision) *selection*. Various weighted cluster aggregation methods have been used in the literature [14], [13], [21]. Unlike these methods, the proposed approach and the algorithms enable learning both the structure of the distance function, the pixel-wise relationships and the *contributions* of the decisions of the segmentation algorithms from the data by solving a single optimization problem using semi-supervision.

In the next section, a brief overview of USF algorithm is given. Semi-supervised Segmentation Fusion algorithm is introduced in Section III. Experimental analyses of the algorithms are given in Section IV. Section V concludes the paper.

II. Unsupervised Segmentation Fusion

In the unsupervised segmentation fusion problem [15], an image $\mathbb{I}$ is fed to J different base-layer segmentation algorithms SA_j, $j = 1, 2, \ldots, J$. Each segmentation algorithm is employed on $\mathbb{I}$ to obtain a set of segmentation outputs $S_j = \{s_i\}_{i=1}^{n_j}$ where $s_i \in A^N$ is a segmentation (partition) output, A is the set of segment labels (names) with N pixels,

$|A| = C$ different segment labels, and a distance function $d(\cdot,\cdot)$. Note that A^N is the class of all segmentations of finite sets with C different segment labels in the image $\mathbb{I}$.

An initial segmentation s is selected from the segmentation set $S = \bigcup_{j=1}^{J} S_j$ consisting of $K = \sum_{j=1}^{J} n_j$ segmentations using algorithms which employ search heuristics, such as Best of K *(BOK)* [11]. Then, a *consensus* segmentation $\hat{s}$ is computed by solving the following optimization problem:

$$\hat{s} = \underset{s}{\operatorname{argmin}} \sum_{i=1}^{K} d(s_i, s) \ .$$

Given two segmentations s_i and s_j, the distance function is defined as the *Symmetric Distance Function (SDD)* given by $d(s_i, s_j) = N_{01} + N_{10}$, where N_{01} is the number of pairs co-segmented in s_i but not in s_j, and N_{10} is the number of pairs co-segmented in s_j but not in s_i [11].

This optimization problem was solved by Ozay et al. [15] using an Unsupervised Segmentation Fusion algorithm. At each iteration t of the optimization algorithm, a new segmentation is computed. Specifically, using the assumption that single element updates do not change the objective function $H_t = \sum_{i=1}^{K} d(s_i, s_t)$, H_t is approximated by H_{t-1} with a scale parameter $\beta \in [0, 1]$. Then, the current *best one element move* is updated at t using

$$\Delta s_t = \frac{\partial}{\partial s_t}(\beta H_{t-1} + d(s_{i'}, s_t)) \ ,$$

where $s_{i'}$ is the randomly selected segmentation. If an $N \times C$ matrix $[H]$ is defined such that the n^{th} row and the c^{th} column of the matrix, $[H]_{nc}$, is the updated value of H obtained by switching n^{th} element of s to the c^{th} segment label, then the move can be approximated by

$$\underset{n,c}{\operatorname{argmin}} \ \beta[H_{t-1}]_{n,c} + [d(s_{i'}, s_t)]_{n,c} \ , \tag{1}$$

if $s_{i'}$ is selected for updating s_t at time t, $\forall i = 1, 2, \ldots, N$, $\forall c = 1, 2, \ldots, C$. If there is no improvement on the best move or a termination time T is achieved, the current segmentation is returned by the USF algorithm [15].

III. Incorporating Prior and Side Information to Segmentation Fusion

In this section, we introduce a new Semi-supervised Segmentation Fusion algorithm which solves weighted decision and distance learning problems that are mentioned in Section I by incorporating side-information about the pixel memberships into the unsupervised Segmentation Fusion algorithm. Then, the goal of the proposed Semi-supervised Segmentation Fusion algorithm can be summarized as obtaining a segmentation which is close to both base-layer segmentations and a target segmentation using weighted distance learning and semi-supervision.

In the weighted distance learning problem, some of the weights may be required to be zero, in other words, sparsity may be required in the space of weight vectors to select the decision of some of the segmentation algorithms. For instance, if fusion is employed on multi-spectral images with large number of bands, and if some of the most informative bands are needed to be selected, then sparsity defined by the weight vectors becomes a very important property. In addition, side information about the pixel-wise relationships of the segmentations can be defined in distance functions. Thereby, both the structure of the distance function, the pixel-wise relationships and the *contributions* of the decisions of the segmentation algorithms can be learned from the data.

A. Formalizing Semi-supervision for Segmentation Fusion

We define Semi-supervised Segmentation Fusion problem as a convex constrained stochastic sparse optimization problem. In the construction of the problem, first pixel-wise segment memberships are encoded in the definition of a semi-supervised weighted distance learning problem by decomposing Symmetric Distance Function (SDD) as [14]

$$d(s_i, s_j) = \sum_{m=1}^{N} \sum_{l=1}^{N} d_{m,l}(s_i, s_j), \tag{2}$$

and

$$d_{m,l}(s_i, s_j) = \begin{cases} 1, & \text{if } (m,l) \in \Theta_c(s_i) \text{ and } (m,l) \notin \Theta_c(s_j) \\ 1, & \text{if } (m,l) \notin \Theta_c(s_i) \text{ and } (m,l) \in \Theta_c(s_j) \\ 0, & \text{otherwise} \end{cases} \ ,$$

where $(m,l) \in \Theta_c(s_i)$ means that the pixels m and l belong to the same segment Θ_c in s_i and $(m,l) \notin \Theta_c(s_i)$ means that m and l belong to different segments in s_i. Then, a connectivity matrix M is defined with the following elements;

$$M_{ml}(s_i) = \begin{cases} 1, & \text{if } (m,l) \in \Theta_c(s_i) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Note that [14],

$$d_{m,l}(s_i, s_j) = [M_{m,l}(s_i) - M_{m,l}(s_j)]^2 \ . \tag{4}$$

Then, the distance between the connectivity matrices of two segmentations s and s_i is defined as [21]

$$d_\kappa(M(s), M(s_i)) = \sum_{m=1}^{N} \sum_{l=1}^{N} d_\kappa(M_{m,l}(s), M_{m,l}(s_i)) \ , \tag{5}$$

where d_κ is the Bregman divergence defined as

$$d_\kappa(x, y) = \kappa(x) - \kappa(y) - \nabla\kappa(y)(x - y) \ ,$$

and $\kappa : \mathbb{R} \to \mathbb{R}$ is a strictly convex function. Since d_κ is defined in (4) as Euclidean distance, (5) is computed during the construction of best one element moves.

In order to compute the weights of base-layer segmentations during the computation of distance functions, the following quadratic optimization problem is defined;

$$\begin{aligned} \underset{\bar{w}}{\operatorname{argmin}} & \sum_{i=1}^{K} w_i d_\kappa(M(s), M(s_i)) + \lambda_q \parallel \bar{w} \parallel_2^2 \\ & \text{s.t.} \sum_{i=1}^{K} w_i = 1, w_i \geq 0, \forall i = 1, 2, \ldots, K \ , \end{aligned} \tag{6}$$

where $\lambda_q > 0$ is the regularization parameter and $\bar{w} = (w_1, w_2, \ldots, w_K)$ is the weight vector. Since we use $\sum_{i=1}^{K} w_i = 1$ and $w_i \geq 0$ in the constraints of the optimization problem (6), we enable the *selection* and *removal* of a base-layer segmentation s_i by assigning $w_i = 0$ to s_i.

Defining the distance function (2) in terms of the segment memberships of the pixels (3) in (4), must-link and cannot-link constraints can be incorporated to the constraints of (6) as follows;

$$M_{ml}(s_i) = \begin{cases} 1, & \text{if}(m,l) \in \mathfrak{M} \\ 0, & \text{if}(m,l) \in \mathfrak{C} \end{cases}, \tag{7}$$

where $\mathfrak{M}$ is the set of must-link constraints and $\mathfrak{C}$ is the set of cannot-link constraints. Then, the following optimization problem is defined for Semi-supervised Segmentation Fusion

$$\begin{aligned} \underset{M(s)}{\operatorname{argmin}} \sum_{i=1}^{K} d_\kappa(M(s), M(s_i)) + \lambda_q \parallel \bar{w} \parallel_2^2 \\ \text{s.t} \quad M_{ml}(s_i) = 1, \text{if}(m,l) \in \mathfrak{M} \\ M_{ml}(s_i) = 0, \text{if}(m,l) \in \mathfrak{C}\,. \end{aligned} \tag{8}$$

Wang, Wang and Li [21] analyze generalized cluster aggregation problem using (8) for fixed weights $\bar{w}$ and define the solution set as follows;

1) If $(m,l) \in \mathfrak{M}$ or $(m,l) \in \mathfrak{C}$, then (7) is the solution set for (k,l),
2) If $(m,l) \notin \mathfrak{M}$ and $(m,l) \notin \mathfrak{C}$, then $M_{ml}(s_i)$ can be solved by

$$\nabla_\kappa M_{ml}(s_i) = \sum_{i=1}^{K} w_i \nabla_\kappa (M(s_i)).$$

Then, they solve (6) for fixed $M(s)$. Note that, ℓ_2 norm regularization does not assure sparsity efficiently [19] because $\parallel \bar{w} \parallel_2^2$ is a quadratic function of the weight variables w_i which treats each w_i equally. In order to control the sparsity of the weights by treating each w_i different from the other weight variables $w_{j \neq i}$ using a linear function of w_i, such as $\parallel \bar{w} \parallel_1$ which is the ℓ_1 norm of $\bar{w}$, a new optimization problem is defined as follows;

$$\begin{aligned} \underset{(M(s),\bar{w})}{\operatorname{argmin}} \sum_{i=1}^{K} w_i d_\kappa(M(s), M(s_i)) + \lambda \parallel \bar{w} \parallel_1 \\ \text{s.t} \quad \sum_{i=1}^{K} w_i = 1, w_i \geq 0, \forall i = 1, 2, \ldots, K \\ M_{ml}(s_i) = 1, \text{if}(m,l) \in \mathfrak{M} \\ M_{ml}(s_i) = 0, \text{if}(m,l) \in \mathfrak{C}\,, \end{aligned} \tag{9}$$

where $\lambda \in \mathbb{R}$ is the parameter which defines the sparsity of $\bar{w}$. Similarly, (9) is computed in two parts;

1) For fixed $M(s)$, solve

$$\begin{aligned} \underset{\bar{w}}{\operatorname{argmin}} \sum_{i=1}^{K} w_i d_\kappa(M(s), M(s_i)) + \lambda \parallel \bar{w} \parallel_1 \\ \text{s.t} \quad \sum_{i=1}^{K} w_i = 1, w_i \geq 0, \forall i = 1, 2, \ldots, K\,. \end{aligned} \tag{10}$$

2) For fixed $\bar{w}$, solve

$$\begin{aligned} \underset{M(s)}{\operatorname{argmin}} \sum_{i=1}^{K} w_i d_\kappa(M(s), M(s_i)) + \lambda \parallel \bar{w} \parallel_1 \\ \text{s.t} \quad M_{ml}(s_i) = 1, \text{if}(m,l) \in \mathfrak{M} \\ M_{ml}(s_i) = 0, \text{if}(m,l) \in \mathfrak{C}\,. \end{aligned} \tag{11}$$

An algorithmic description of Semi-supervised Segmentation Fusion which solves (10) and (11) is given in the next subsection.

B. Semi-supervised Segmentation Fusion Algorithm

In the proposed Semi-supervised Segmentation Fusion algorithm, (10) and (11) are solved to compute weighted distance functions which are used in the construction of best one element moves.

In Algorithm 1, first the weight vector $\bar{w}$ is computed by solving (10) for each selected segmentation $s_{i'}$ in the 4^{th} step of the algorithm. In order to solve (10) using an optimization method called Alternating Direction Method of Multipliers (ADMM) [3]. ADMM has been employed to solve (10) until a termination criterion $\tau \leq T_\tau$ or convergence is achieved [3]. Once the weight vector $\bar{w}$ is computed in the 4^{th} step, (11) is solved in the 5^{th}, 6^{th} and 7^{th} steps of the algorithm: $\bar{w}d(s_{i'}, s) + \lambda \parallel \bar{w} \parallel_1$ is computed using $M(s_{i'})$ and $\bar{w}$ in the 5^{th} step, $[H_t]$ is computed in the 6^{th} step and Δs is computed in the 7^{th} step to update s. Note that the sparse weighted distance function, which is approximated by $\beta[H_t] + [\bar{w}d(s_{i'}, s) + \lambda \parallel \bar{w} \parallel_1]$ in Algorithm 1, is different from the distance function in USF.

In addition, each segmentation is selected sequentially in a pseudo-randomized permutation order in Algorithm 1. If an initially selected segmentation performs better than the other segmentations, then the algorithm may be terminated in the first running over the permutation set. Otherwise, the algorithm runs until the termination time T is achieved or all of the segmentations are selected.

input : Input image I, $\{SA_j\}_{j=1}^{J}$, T, T_τ.
output: Output segmentation O.
1 Run SA_j on I to obtain $S_j = \{s_i\}_{i=1}^{u_j}$, $\forall j = 1, 2, \ldots, J$.;
2 At $t = 1$, initialize s and $[H_t]$;
for $t \leftarrow 2$ **to** T **do**
3 Randomly select one of the segmentation results with an index $i' \in \{1, 2, \ldots, K\}$;
4 Solve (10) for $M(s_{i'})$ to compute w_k;
5 Compute $\bar{w}d(s_{i'}, s) + \lambda \parallel \bar{w} \parallel_1$;
6 $[H_t] \leftarrow \beta[H_t] + [\bar{w}d(s_{i'}, s) + \lambda \parallel \bar{w} \parallel_1]$;
7 Compute Δs by solving $\underset{n,c}{\operatorname{argmin}} \beta[H_t]_{n,c}$;
8 $s \leftarrow s + \Delta s$;
9 $t \leftarrow t + 1$;
end
10 $O \leftarrow s$;

Algorithm 1: Semi-supervised Segmentation Fusion.

IV. Experiments

In this section, the proposed Semi-supervised Segmentation Fusion (SSSF) algorithm is analyzed on real world benchmark multi-spectral and aerial images [22], [16], [2]. In the implementations, three well-known segmentation algorithms, k-means, Mean Shift citems and Graph Cuts [4], [1], [5] are used as the base-layer segmentation algorithms. Three indices are used to measure the performances between the output images O and the ground truth of the images: i) Rand Index (RI), ii) Adjusted Rand Index (ARI), and iii) Adjusted Mutual Information (AMI) [20] which adjusts the effect of mutual information between segmentations due to chance, similar to the way the ARI corrects the RI.

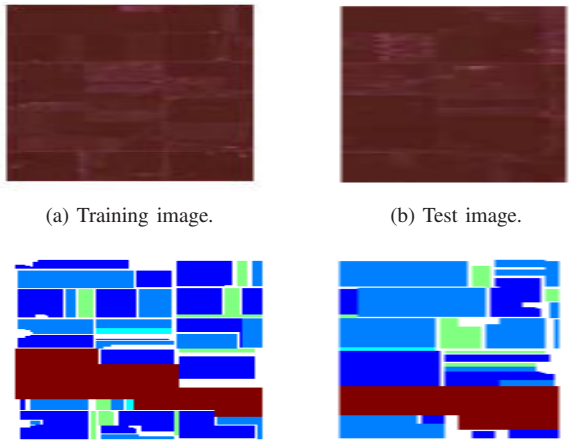
In the experiments, a Graph Cut implementation of Veksler [5] for image segmentation is used with a Matlab wrapper of Bagon [1] and the source code provided by Shi [18]. The algorithm parameters are selected by first computing ARI values between a given target segmentation and each segmentation computed for each parameter $\sigma_r \in \{0.1, 0.2, \ldots, 10\}$, $\sigma_s \in \{1, 2, \ldots, 100\}$, $r_{ncut} \in \{1, 2, \ldots, 100\}$ and $\tau_{ncut} \in \{0.01, 0.02, \ldots, 1\}$ [18]. Then, a parameter 4-tuple $(\sigma_r, \sigma_s, r_{ncut}, \tau_{ncut})$ which maximizes ARI is selected[1]. Similarly, a parameter 3-tuple (h_s, h_r, mA) which maximizes ARI is selected for Mean Shift algorithm from the parameter sets $h_s \in \{1, 3, 5, 10, 50, 100\}$, $h_r \in \{1, 3, 5, 10, 50, 100\}$ and $mA \in \{100, 200 \ldots, 10000\}$ [7]. For k-means, $k = C$ is used, if not stated otherwise. Assuming that C is not known in the image, a parameter search algorithm proposed in [15] is employed using the training data in order to find the optimal $\hat{C}$ for $c = 2, 3, 4, 5, 6, 7, 8, 9, 10$. Similarly, the parameter estimation algorithm suggested in [15] is employed for a set of $\hat{\beta}$ values $\Xi = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99\}$.

The termination parameter of SSSF and ADMM is taken as $T = 1000$ and $T_\tau = 1000$, respectively. The penalty parameter of ADMM is chosen as $\theta = 1$ as suggested in [3]. The regularization parameter is computed as $\lambda = 0.5\lambda_{max}$ [3], where $\lambda_{max} = max\{\|d_\kappa(M(s), M(s_a))\bar{y}\|_2\}_{a=1}^{K}$, $y_n = \|d_\kappa(M(s), M(s_n))\bar{w}\|_2$, $S = \{s_n\}_{n=1}^{N}$ is the set of segments in an training image and $\bar{y} = [y_1, y_2, \ldots, y_N]$ is the labels of segments in S. Then, λ is computed in training phase and employed in both training and test phases. In the training phase, λ and $\bar{w}$ are computed, and the constraints $\mathfrak{M}$ and $\mathfrak{C}$ are constructed using the ground truth data, i.e. pixel labels of training images as described in Section III-A. In the testing phase, (3) is employed for the construction of connectivity matrices and $[\bar{w}d(s_i k, s) + \lambda \parallel \bar{w} \parallel_1]$ is computed $\forall i = 1, 2, \ldots, K$. The performance of the proposed SSSF is compared with the performances of k-means, Mean Shift, Graph Cuts, Unsupervised Segmentation Fusion (USF) [15], Distance Learning (DL) [15] and Quasi-distance Learning (QD) [15] algorithms.

[1]Minimization of ARI is considered as the cost function in the estimation of parameters following the relationship between ARI and SDD as well as it is one of the performance measures [15].

A. Analyses on Multi-spectral Images

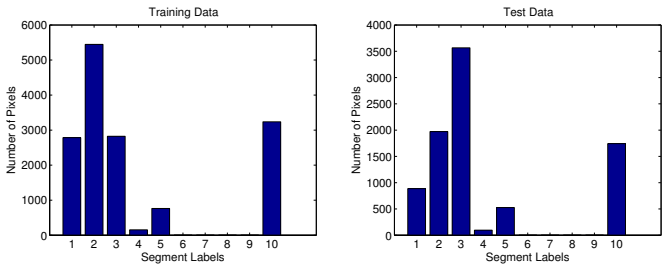
In the first set of experiments, the proposed algorithms are employed on 7 band Thematic Mapper Image (TMI) which is provided by MultiSpec [2]. The image with size 169×169 is split into training and test images: i) a subset of the pixels with coordinates $x = (1 : 169)$ and $y = (1 : 90)$ is taken as the training image and ii) a subset of the pixels with coordinates $x = (1 : 169)$ and $y = (91 : 142)$ is taken as the test image. Dataset is split in order to obtain segments with at least 100 pixels both in training and test images. Training and test images are shown in Figure 1 with their Ground Truth (GT) labels. In the images, there are $C = 6$ number of different segment labels. The distribution of pixels given the segment labels is shown in Figure 2.



(a) Training image. (b) Test image.

(c) GT of training image. (d) GT of test image.

Fig. 1: Training and test images obtained from TMI.



(a) Training dataset. (b) Test dataset.

Fig. 2: Distribution of pixels given the segment labels in TMI.

First k-means is implemented on different bands $\mathbb{I}_j$ of the multi-spectral image $\mathbb{I} = (\mathbb{I}_1, \mathbb{I}_2, \ldots, \mathbb{I}_J)$ for $J = 7$, in order to perform multi-modal data fusion of different spectral bands using segmentation fusion. The results of the experiments on Thematic Mapper Image is given in Table I. In the Average Base column, the performance values of k-means algorithm averaged over 7 bands are given. It is observed that the performance values of USF are similar to the arithmetic average

TABLE I: Training and test performances of the algorithms for Thematic Mapper Image.

	Average Base		USF		DL		QD		SSSF	
	Tr	Te	Tr	Te	Tr	Te	Tr	Te	Tr	Te
RI	0.730	0.703	0.731	0.704	0.738	0.710	0.732	0.714	0.792	0.740
ARI	0.264	0.159	0.265	0.160	0.282	0.184	0.270	0.174	0.305	0.220
AMI	0.182	0.187	0.182	0.188	0.205	0.203	0.198	0.204	0.251	0.237

TABLE II: Experiments on 7-band images.

	k-means		Graph Cut		Mean Shift		USF		DL		QD		SSSF	
	Tr	Te	Tr	Te	Tr	Te	Tr	Te	Tr	Te	Tr	Te	Tr	Te
RI	0.742	0.715	0.754	0.717	0.710	0.714	0.711	0.714	0.713	0.710	0.752	0.724	0.801	0.733
ARI	0.167	0.125	0.234	0.132	0.266	0.176	0.267	0.176	0.270	0.180	0.262	0.178	0.326	0.236
AMI	0.176	0.183	0.193	0.190	0.195	0.209	0.196	0.209	0.195	0.205	0.198	0.211	0.220	0.219

TABLE III: Performance of the algorithms for Moderate Dimension Image.

	Average Base		USF		DL		QD		SSSF	
	Tr	Te	Tr	Te	Tr	Te	Tr	Te	Tr	Te
RI	0.533	0.532	0.532	0.530	0.533	0.533	0.535	0.530	0.553	0.550
ARI	0.008	0.009	0.007	0.007	0.013	0.011	0.010	0.011	0.109	0.110
AMI	0.139	0.141	0.124	0.120	0.123	0.121	0.123	0.124	0.177	0.185

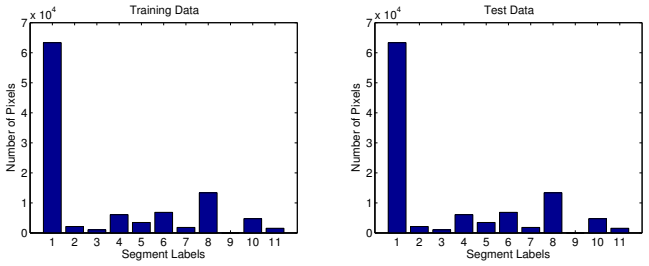
TABLE IV: Performances of algorithms on Road Segmentation Dataset.

	k-means		Graph Cut		Mean Shift		USF		DL		QD		SSSF	
	Tr	Te	Tr	Te	Tr	Te	Tr	Te	Tr	Te	Tr	Te	Tr	Te
RI	0.513	0.535	0.512	0.523	0.379	0.328	0.378	0.328	0.392	0.353	0.407	0.390	0.550	0.563
ARI	0.014	0.002	0.017	0.008	0.010	0.008	0.010	0.008	0.010	0.008	0.011	0.007	0.020	0.015
AMI	0.404	0.003	0.054	0.006	0.053	0.070	0.044	0.070	0.082	0.080	0.090	0.080	0.422	0.110

of the performance values of k-means algorithms. When semi-supervision is used, a remarkable increase is observed in the performances in SSSF. However, full performance (1 values for the indices) is not achieved in training. Since the output image O may not converge to the GT of the image, the convergence assumption mentioned in the previous section may not be valid for this image.

In the second set of the experiments, k-means, Graph Cut and Mean Shift algorithms are employed on 7-band training and test images. Now, the image segmentation problem is considered as a pixel clustering problem in 7 dimensional spaces. The results are given in Table II. The performance values of USF are closer to the performance values of the Mean Shift algorithm, since the output image of USF is closer to the output segmentation of the Mean Shift algorithm. Moreover, SSSF provides better performance than the other algorithms, since SSSF incorporate prior information by assigning higher weights to the partitions with higher performances.

In the third set of experiments, k-means algorithm is employed on each band of 12-band Moderate Dimension Image [2]. The size of the image is 949×220, and there are 11 segments in the GT of the image [2]. The classes are background, Alfalfa, Br Soil, Corn, Oats, Red Cl, Rye, Soybeans, Water, Wheat, Wheat2. 104392 pixels are randomly selected for training and the remaining 104388 pixels are randomly selected for testing. In order to conserve the spatial distribution of the selected pixels, the pixels which reside in a segment with the same label in a spatial neighborhood are selected as test and training data. The distributions of pixels in training and test datasets are shown in Figure 3. The results on the test data are given in Table III. It is observed that the performance values for USF are smaller than the average performance values of base-layer segmentation outputs. When prior information is employed using SSSF, it is observed that the smaller weights are assigned to the segmentations with relatively small performance values. In addition, the output images of SSSF are closer to the target segmentations obtained from the GT images. In summary, remarkable performance increases are observed in SSSF algorithm.



(a) Training dataset. (b) Test dataset.

Fig. 3: Distribution of pixels given the segment labels in MDI.

B. Analyses on Aerial Images

In this section, the segmentation of roads in the aerial images is considered, which are analyzed in [22]. Detailed information about the images in the dataset is given in [22], [16].

7 training and 7 test images with road and background labels are randomly selected from the dataset. The id numbers

of the training and test images in the dataset are tr = $\{7, 26, 40, 41, 42, 43, 77\}$, and te = $\{78, 90, 91, 92, 93, 94, 95\}$, respectively. In order to observe the affect of the statistical similarity between training and test datasets, the performances are not averaged for different implementations of algorithms on random permutations of training and test images, and both of training and test performances are given in the results.

The results are shown in Table IV. It is observed that the performance indices of USF are the same as the indices of Mean Shift. This is basically because of the fact that Mean Shift has a higher number of different segment labels than the other algorithms. Therefore, the outputs of Mean Shift suppress the outputs of other algorithms in the computation of distance functions. Moreover, higher performances than the base-layer segmentation algorithms are obtained, when semi-supervision (SSSF) is employed in segmentation fusion.

V. Conclusion

An algorithm called Semi-supervised Segmentation Fusion (SSSF) is introduced for fusing the segmentation outputs (decisions) of base-layer segmentation algorithms by incorporating the prior information about the data statistics and side-information about the content into the Unsupervised Segmentation Fusion algorithm. The proposed SSSF algorithm reformulates the segmentation fusion problem as a constrained optimization problem, where the constraints are defined in such a way to semi-supervise the segmentation process.

Experimental results show that the difference between RI and ARI values increases, as the number of segmentation outputs K increases for a fixed number of segments C. We observe that one of the reasons for the observation of this fluctuation is the early termination of the USF and the proposed SSSF before a consensus segmentation is obtained. In addition, the performances of the base-layer segmentation algorithms and the proposed segmentation fusion algorithms are sensitive to the statistical similarity of the images used in training and test datasets. The sensitivity of the base-layer segmentation algorithms affect the performances of the USF algorithm. Moreover, the employment of semi-supervision on the USF using Semi-supervised Segmentation Fusion algorithm further increase the performances.

Note that the performances of the proposed algorithms can be improved by the theoretical analyses on their open problems such as the investigation and modeling the dependency of the performances on the algorithm parameters, the statistical properties of the segmentations and images in training and test datasets, which are postponed to the future work.

Acknowledgement

This work was supported by the European commission project PaCMan EU FP7-ICT, 600918.

References

[1] S. Bagon, "Matlab wrapper for graph cut," Dec 2006. [Online]. Available: http://www.wisdom.weizmann.ac.il/~bagon

[2] L. Biehl and D. Landgrebe, "Multispec: a tool for multispectral–hyperspectral image data analysis," *Comput and Geosci*, vol. 28, pp. 1153–1159, Dec 2002.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan 2011.

[4] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *Int J Comput Vision*, vol. 70, no. 2, pp. 109–131, Nov 2006.

[5] Y. Boykov, O. Veksler, and R. Zabih, "Efficient approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1222–1239, Nov 2001.

[6] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

[7] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[8] B. Dasarathy, *Decision fusion*. IEEE Computer Society Press, 1994.

[9] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[10] L. Franek, D. D. Abdala, S. Vega-Pons, and X. Jiang, "Image segmentation fusion using general ensemble clustering methods," in *Proceedings of ACCV'10*, ser. ACCV'10, 2011, pp. 373–384.

[11] A. Goder and V. Filkov, "Consensus clustering algorithms: Comparison and refinement," in *Proc. SIAM Workshop on Algorithm Engineering and Experiments*, J. I. Munro and D. Wagner, Eds., 2008, pp. 109–117.

[12] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2001.

[13] T. Li, C. Ding, and M. I. Jordan, "Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization," in *Proc. Int. Conf. Machine Learning (ICDM ',07)*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 577–582.

[14] T. Li and C. H. Q. Ding, "Weighted consensus clustering," in *SIAM Int. Conf. on Data Mining*, Atlanta, Georgia, 2008, pp. 798–809.

[15] M. Ozay, F. Yarman Vural, S. Kulkarni, and H. Poor, "Fusion of hyperspectral image segmentation algorithms using consensus clustering," in *Proc of Int Conf Image Processing, (ICIP 2013)*, Sep 2013.

[16] J. Porway, Q. Wang, and S. C. Zhu, "A hierarchical and contextual model for aerial image parsing," *Int J Comput Vision*, vol. 88, no. 2, pp. 254–283, Jun 2010.

[17] V. Sharma and J. Davis, "Feature-level fusion for object segmentation using mutual information," in *Augmented Vision Perception in Infrared*, ser. Advances in Pattern Recognition, R. Hammoud, Ed. Springer London, 2009, pp. 295–320.

[18] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 888–905, Aug 2000.

[19] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J Roy Stat Soc B*, vol. 58, pp. 267–288, 1996.

[20] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: is a correction for chance necessary?" in *Proc of Int Conf Machine Learning (ICML)*, 2009, pp. 1073–1080.

[21] F. Wang, X. Wang, and T. Li, "Generalized cluster aggregation," in *Proc of IJCAI*, 2009, pp. 1279–1284.

[22] B. Yao, X. Yang, and S.-C. Zhu, "Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks," in *Proc. Int. Conf. Energy Minimization Comput. Vis. Pattern Recognit.*, 2007, pp. 169–183.

[23] H. Zhang, J. E. Fritts, and S. A. Goldman, "Image segmentation evaluation: A survey of unsupervised methods," *Comput Vis Image Und*, vol. 110, no. 2, pp. 260–280, 2008.

A New Fuzzy Stacked Generalization Technique and Analysis of its Performance

Mete Ozay, *Member, IEEE,* Fatos T. Yarman Vural, *Senior Member, IEEE*

Abstract

A new Stacked Generalization method which employs a hierarchical distance learning strategy in a two-layer ensemble learning architecture called Fuzzy Stacked Generalization (FSG) is proposed. At the base-layer of FSG, fuzzy k-Nearest Neighbor (k-NN) classifiers map their own input feature vectors into the posteriori probabilities. At the meta-layer, a fuzzy k-NN classifier learns a distance function by minimizing the difference between the large sample and N-sample classification error using the estimated posteriori probabilities. In the FSG, the feature space of each base-layer classifier is designed to gain an expertise on a specific property of the dataset, whereas the meta-layer classifier learns the degree of accuracy of the decisions of the base-layer classifiers. Experimental results obtained using the artificial datasets show that the classification performance of the FSG depends on diversity and cooperation of the classifiers rather than the classification performances of the individual base-layer classifiers. A weak base-layer classifier may boost the overall performance of the FSG more than a strong classifier, if it is capable of recognizing the samples, which are not recognized by the rest of the classifiers. The cooperation among the base-layer classifiers is quantified by introducing a *shearability* measure. The effect of the shearability on the performance is investigated on the artificial datasets. Experiments on the real datasets show that FSG performs better than the state of the art ensemble learning algorithms such as, Adaboost, Random Subspace and Rotation Forest.

Index Terms

Fuzzy classification, nearest neighbor rule, hierarchical decision fusion, distance learning.

M. Ozay is with the School of Computer Science, University of Birmingham, Edgbaston, Birmingham, United Kingdom. F. T. Yarman Vural is with the Department of Computer Engineering, Middle East Technical University, Ankara, Turkey. E-mail: m.ozay@cs.bham.ac.uk, vural@ceng.metu.edu.tr.

I. INTRODUCTION

Stacked Generalization algorithm, proposed by Wolpert [1] and used by many others [2], [3], [4], [5], [6], [7] is a widely used ensemble learning technique. The basic idea is to ensemble several classifiers in various ways so that the performance of the Stacked Generalization (SG) is higher than that of the individual classifiers which take place under the ensemble. Although gathering the classifiers under the Stacked Generalization algorithm significantly boosts the performance in some application domains, it is observed that the performance of the overall system may get worse than that of the individual classifiers in some other cases. Wolpert defines the problem of describing the relation between the performance and various parameters of the algorithm as a black art problem [1], [7].

In this study, we suggest a Fuzzy Stacked Generalization (FSG) method and resolve the black art problem [1] by minimizing the difference between the large sample and N-sample classification errors. The proposed technique aggregates the independent decisions of the fuzzy k-Nearest Neighbor (k-NN) classifiers in the ensemble. A meta-layer fuzzy classifier is, then, trained to learn the degree of *correctness* and *expertise* of the base-layer classifiers.

There are three major contributions of this study:

1) A novel hierarchical distance learning approach, which minimizes the difference between N-sample and large-sample classification error of the nearest neighbor algorithm, is proposed.
2) In the proposed FSG, a specific feature space is designed for each base-layer classifier. This approach enables us to create expert base-layer classifiers each of which is trained to learn a particular property of the sample set. The expert classifiers are then trained to collaborate in order to correctly label the samples in the FSG architecture.
3) The black art problem of the FSG is empirically analyzed and *contribution* of each base-layer classifier on the overall performance is investigated. It is observed that if the base-layer classifiers share all the samples in the training set to correctly classify them, then the performance of the overall FSG becomes higher than that of the individual base-layer classifiers. On the other hand, if a sample is misclassified by all of the base-layer classifiers, then this sample causes a performance decrease of the FSG.

The suggested Fuzzy Stacked Generalization algorithm is tested on artificial and real datasets, and compared with the state of the art ensemble learning algorithms such as Adaboost [8], Random Subspace [9] and Rotation Forest [10].

In the next section, our motivation is given with a brief literature review. Distance learning problem for a single classifier is defined in Section III, and extended into ensemble of classifiers in Section IV. Employment of the proposed distance learning approach to stacked generalization method is given in Section V. Computational complexity of the proposed FSG is analyzed in Section VI. Experimental analyses are given in Section VII. Section VIII concludes the paper.

II. Related Work and Motivation

Among a wide range of Stacked Generalization methods, we suffice to review the ones which are similar to the suggested FSG architecture, where the decisions of the ensemble of classifiers are fused by the vector concatenation operation [1], [2], [3], [4], [5], [6], [11], [12], [13], [14], [15], [16], [17]. As a good example, Ueda aggregates the decisions of an ensemble of Neural Networks by vector concatenation operation and compares his method to the voting methods in an experimental setup [2]. Following the same formulation, Sen and Erdogan [3] analyze various weighted and sparse linear combination methods by combining decisions of heterogeneous base-layer classifiers, such as decision trees and k-NN method. In another study, Rooney et al. [4] employ homogeneous and heterogeneous classifier ensembles for stacked regression using linear combination rules. Similarly, Yarman Vural et al. [13], [14], [16] suggest several homogeneous SG algorithms using fuzzy k-NN classifiers and compare the classification performance of their method to popular ensemble learning methods. A comparative study is done by Zenko et al. [5], who employ linear combination rules with the ensemble learning algorithms, such as bagging, boosting and voting. Sigletos et al. [15] compare the classification performances of several SG algorithms which combine the crisp decision values and/or probabilistic decisions.

Performance evaluations of the stacked generalization methods reported in the literature are not consistent with each other. This fact is demonstrated by Dzeroski and Zenko in [17] where they report contradicting results with the studies in the literature on SG. The contradictory

results can be attributed to many non-linear relations among the parameters of the SG, such as the number of classifiers and their feature spaces.

Designing the feature spaces and classifiers, which boost the performance of an SG method, has been considered as a black art problem by Wolpert [1], and Ting and Witten [7]. Most of the time, popular classifiers, such as k-NN, Neural Networks and Naive Bayes, are used as the base-layer classifiers in SG. However, due to numerous nonlinear relations and incompatibilities among the parameters, tracing the feature mappings from base-layer input feature spaces to meta-layer output decision space becomes an intractable and uncontrollable problem. Additionally, the heterogeneous classifiers generate different type of information about the decisions, such as crisp, fuzzy or probabilistic class labellings.

The employment of fuzzy decisions in the ensemble learning algorithms is analyzed in [6], [18], [19]. Tan et al. [6] use fuzzy k-NN algorithms at the base-layer classifiers and employ a linearly weighted voting method to combine the fuzzy decisions. Cho and Kim [18] combine the decisions of Neural Networks using a fuzzy combination rule called fuzzy integral. Kuncheva [19] experimentally compares various fuzzy and crisp combination methods, including fuzzy integral [20] and voting, to boost the classifier performances in Adaboost. In her experimental results, the classification algorithms that implement fuzzy rules outperform the algorithms that implement crisp rules. However, the effect of the employment of fuzzy rules to the classification performance of SG is mentioned as an open problem.

In this study, most of the above mentioned intractable problems are avoided by designing a homogeneous Stacked Generalization model, called Fuzzy Stacked Generalization (FSG). This model consists of a set of base-layer classifiers each of which extracts complementary information from the feature vectors of each sample residing in a different feature space. The fuzzy k-NN classifiers of the base-layer are considered as feature mappings from the feature vectors of the input space to posteriori probabilities of decision space. The fuzzy k-NN classifiers, also, enable us to obtain information about the uncertainty of the classifier decisions and the belongingness of the samples to classes [20], [21], [22]. A meta-layer classifier is then designed to learn the degree of expertise of each base-layer classifier. This task is achieved by formulating the classification error of the proposed FSG in two parts, namely i) N-sample error which is the error of a classifier employed on a training dataset of

N samples and ii) large-sample error which is the error of a classifier employed on a training dataset of large number of samples such that $N \rightarrow \infty$. A distance learning approach proposed by Short and Fukunaga [23] is extended into hierarchical FSG architecture for decision fusion in order to minimize the difference between N-sample and large-sample error.

In the literature, distance learning methods have been employed to prototype and feature selection [24], [25], [26], [27], [28] and weighting [29] methods by computing the weights associated to samples and feature vectors, respectively. The computed weights are used to transform feature spaces of classifiers to more discriminative spaces [30], [31], [32] in order to decrease the N-sample classification error of the classifiers [33]. A detailed literature review of prototype selection and distance learning methods for nearest neighbor classification is given in [27].

III. N-SAMPLE AND LARGE-SAMPLE CLASSIFICATION ERRORS OF A SINGLE k-NN CLASSIFIER

In this section, we first define the large sample and N-sample classification errors of a k-NN classifier. Then, we minimize the expected square of the difference between the large sample and N-sample classification errors by designing the distance function employed in the k-NN classifier.

Suppose that a training dataset $S = \{(s_i, y_i)\}_{i=1}^{N}$ of N samples, where $y_i \in \{\omega_c\}_{c=1}^{C}$ is the label of a sample s_i is given. A sample s_i is represented in a feature space F by a feature vector $\bar{x}_i \in \mathbb{R}^D$.

Given a new test sample (s'_i, y'_i) with $\bar{x}'_i \in F$, the nearest neighbor rule (e.g. $k = 1$) simply estimates the label of $\bar{x}'_i$ as the label of the nearest neighbor of $\bar{x}'_i$. In the k-Nearest Neighbor rule (e.g. k-NN), $\hat{y}'_i$ is estimated as

$$\hat{y}'_i = \arg\max_{\omega_c} \mathcal{N}(\eta_k(\bar{x}'_i), \omega_c),$$

where $\mathcal{N}(\eta_k(\bar{x}'_i), \omega_c)$ is the number of samples which belong to ω_c in a neighborhood system $\eta_k(\bar{x}'_i)$. Then the probability of error $\epsilon(\bar{x}_i, \bar{x}'_i) = P_N(error|\bar{x}_i, \bar{x}'_i)$ of the nearest neighbor rule

is computed using N number of samples as

$$\epsilon(\bar{x}_i, \bar{x}'_i) = 1 - \sum_{c=1}^{C} P(\omega_c|\bar{x}_i)P(\omega_c|\bar{x}'_i), \tag{1}$$

where $P(\omega_c|\bar{x}_i)$ and $P(\omega_c|\bar{x}'_i)$ represent posterior probabilities for ω_c [34].

In the asymptotic of large number of training samples, if $P(\omega_c|\bar{x}_i)$ is continuous at $\bar{x}'_i$, then large-sample error $\epsilon(\bar{x}'_i) = \lim_{N\to\infty} P_N(error|\bar{x}'_i)$ is computed as

$$\epsilon(\bar{x}'_i) = 1 - \sum_{c=1}^{C} P^2(\omega_c|\bar{x}'_i). \tag{2}$$

Therefore, the difference between the N-sample error (1) and the large-sample error (2) is computed as

$$\epsilon(\bar{x}_i, \bar{x}'_i) - \epsilon(\bar{x}'_i) = \sum_{c=1}^{C} (P(\omega_c|\bar{x}'_i))(P(\omega_c|\bar{x}_i) - P(\omega_c|\bar{x}'_i)). \tag{3}$$

There is an elegant relationship between the errors of Bayes classifier ($e*$), and N-sample and large-sample errors of k-NN as follows [35]:

$$e* \leq \epsilon(\bar{x}'_i) \leq \epsilon(\bar{x}_i, \bar{x}'_i) \leq 2e*.$$

Note that, if k grows with N, such that $\lim_{N\to\infty} k \to \infty$ as $\lim_{N\to\infty} \frac{k}{N} \to 0$, then the classification error of k-NN converges to that of Bayes classifier [35], [36]. Therefore, the minimization of $E_N\{\big(\epsilon(\bar{x}_i, \bar{x}'_i) - \epsilon(\bar{x}'_i)\big)^2\}$, where the expectation is computed over the number of training samples N, enables us to get closer to the Bayes error ($e*$).

Short and Fukunaga [23] show that $E_N\{\big(\epsilon(\bar{x}_i, \bar{x}'_i) - \epsilon(\bar{x}'_i)\big)^2\}$ can be minimized by either increasing N or designing a distance function $d(\bar{x}'_{i,j}, \cdot)$ which will be employed for the computation of the neighborhood system of the classifier. In a classification problem, an appropriate distance function is computed as [23]

$$d(\bar{x}'_i, \bar{x}_i) = \|\bar{P}(\bar{x}_i) - \bar{P}(\bar{x}'_i)\|_2^2, \tag{4}$$

where $\bar{P}(\bar{x}_i) = [P(\omega_c|\bar{x}_i)]_{c=1}^C$ which is defined as $[P(\omega_c|\bar{x}_i)]_{c=1}^C = [P(\omega_1|\bar{x}_i), \ldots, P(\omega_C|\bar{x}_i)]$, $\bar{P}(\bar{x}'_i) = [P(\omega_c|\bar{x}'_i)]_{c=1}^C$ which is defined as $[P(\omega_c|\bar{x}'_i)]_{c=1}^C = [P(\omega_1|\bar{x}'_i), \ldots, P(\omega_C|\bar{x}'_i)]$ and $\|\cdot\|_2^2$ is the squared ℓ_2 norm.

The main goal of this paper is to design an ensemble learning architecture which minimizes the difference between N-sample and large-sample errors. For this purpose, first (3) is extended to the case where there is an ensemble of classifiers. Then a hierarchical architecture, called Fuzzy Stacked Generalization, is proposed to minimize this error difference by employing a distance learning approach suggested by Short and Fukunaga, as explained in Sections IV and V.

IV. N-SAMPLE AND LARGE-SAMPLE CLASSIFICATION ERROR DIFFERENCE IN ENSEMBLE OF CLASSIFIERS

Suppose that J different features are extracted from each sample $s_i \in S$. Each feature is represented in a feature space F_j by a feature vector $\bar{x}_{i,j} \in \mathbb{R}^{D_j}$, $\forall j = 1, 2, \ldots, J$. The feature vectors residing at space F_j are fed to a classifier $\Gamma_j, \forall j = 1, 2, \ldots, J$. Then, we define a difference function between the large sample and N-sample errors for each classifier Γ_j and for each class ω_c as

$$e_c(\bar{x}_{i,j}, \bar{x}'_{i,j}) = \big(P(\omega_c|\bar{x}_{i,j}) - P(\omega_c|\bar{x}'_{i,j})\big)^2$$

and an overall error function for each classifier Γ_j as $e(\bar{x}_{i,j}, \bar{x}'_{i,j}) = \sum_{c=1}^{C} \epsilon_c(\bar{x}_{i,j}, \bar{x}'_{i,j})$ for a given test sample $\bar{x}'_{i,j} \in F_j$. Therefore, for each Γ_j in the ensemble, we need to minimize

$$E_N\{e^2(\bar{x}_{i,j}, \bar{x}'_{i,j})\}, \tag{5}$$

where the expectation is computed over the number of training samples N. Note that, according to [23], minimization of (5) is equivalent to minimization of the expected square of (3).

If the N-sample error is minimized on each feature space $F_j, \forall j = 1, 2, \ldots, J$, then an average error over an ensemble of classifiers $\hat{E}_J\{E_N\{e^2(\bar{x}_{i,j}, \bar{x}'_{i,j})\}\}$ which is defined as

$$\hat{E}_J\{E_N\{e^2(\bar{x}_{i,j}, \bar{x}'_{i,j})\}\} = \frac{1}{J}\sum_{j=1}^{J} E_N\{e^2(\bar{x}_{i,j}, \bar{x}'_{i,j})\} \tag{6}$$

is minimized by minimizing the following distance function

$$d(s'_i, s_i) = \sum_{j=1}^{J} \|\bar{P}(\bar{x}_{i,j}) - \bar{P}(\bar{x}'_{i,j})\|_2^2, \tag{7}$$

where $\bar{P}(\bar{x}_{i,j}) = [P(\omega_c|\bar{x}_{i,j})]_{c=1}^{C}$ and $\bar{P}(\bar{x}'_{i,j}) = \left[P(\omega_c|\bar{x}'_{i,j})\right]_{c=1}^{C}$.

The right hand side of (7) consists of the posteriori probabilities obtained at the output of classifiers in the ensemble. The classifiers can be fused in such a way that the distance function $d(s'_i, s_i)$ becomes minimum. The following section presents a hierarchical ensemble learning architecture, called Fuzzy Stacked Generalization, which minimizes the distance function of (7).

V. Fuzzy Stacked Generalization for Hierarchical Distance Learning

The suggested Fuzzy Stacked Generalization (FSG) architecture has two layers: The first layer, called base-layer, consists of ensemble of classifiers which are employed to estimate the *posterior* probabilities for each input feature space. In the second layer, called meta-layer, the distance function of (7) is minimized and class labels of test samples are predicted. Flowchart of the architecture is shown in Fig. 1, and explained in detail in this section.

Base-Layer: At the base-layer of the FSG, each fuzzy k-NN classifier Γ_j receives a set of feature vectors $\{\bar{x}_{i,j}\}_{i=1}^{N}$, where $\bar{x}_{i,j} \in F_j$ is extracted from a sample s_i obtained from a training dataset $S = \{(s_i, y_i)\}_{i=1}^{N}$ using a feature extraction algorithm FE_j, $\forall j = 1, 2, \ldots, J$ (see Fig. 1). The output of a fuzzy k-NN classifier, Γ_j, is a set of fuzzy class membership values $\mu_c(\bar{x}_{i,j})$ which are computed by

$$\mu_c(\bar{x}_{i,j}) = \frac{\sum_{n=1}^{k} y_{l(n)} \left(\left\|\bar{x}_{i,j} - \bar{x}_{l(n),j}\right\|_2\right)^{-\frac{2}{\varphi-1}}}{\sum_{n=1}^{k} \left(\left\|\bar{x}_{i,j} - \bar{x}_{l(n),j}\right\|_2\right)^{-\frac{2}{\varphi-1}}}, \tag{8}$$

where $y_{l(n)}$ is the label of the n^{th}-nearest neighbor $x_{l(n),j}$ of $\bar{x}_{i,j}$, and φ is the fuzzification parameter [37], $\forall c = 1, 2, \ldots, C$, $\forall i = 1, 2, \ldots, N$, $\forall j = 1, 2, \ldots, J$. Then the *posteriori* probabilities are approximated by the class membership values of each base-layer classifier, i.e.,

$$P(\omega_c|\bar{x}_{i,j}) \approx \mu_c(\bar{x}_{i,j}). \tag{9}$$

In the training step, the class membership value $\mu_c(\bar{x}_{i,j})$ of each sample s_i is computed by leave-one-out cross validation for each $(\bar{x}_{i,j}, y_i)$ in the validation set $S_j^{CV} = S_j - (\bar{x}_{i,j}, y_i)$, where $S_j = \{(\bar{x}_{n,j}, y_n)\}_{n=1}^{N}$. The class label of an unknown sample s_i is estimated by a

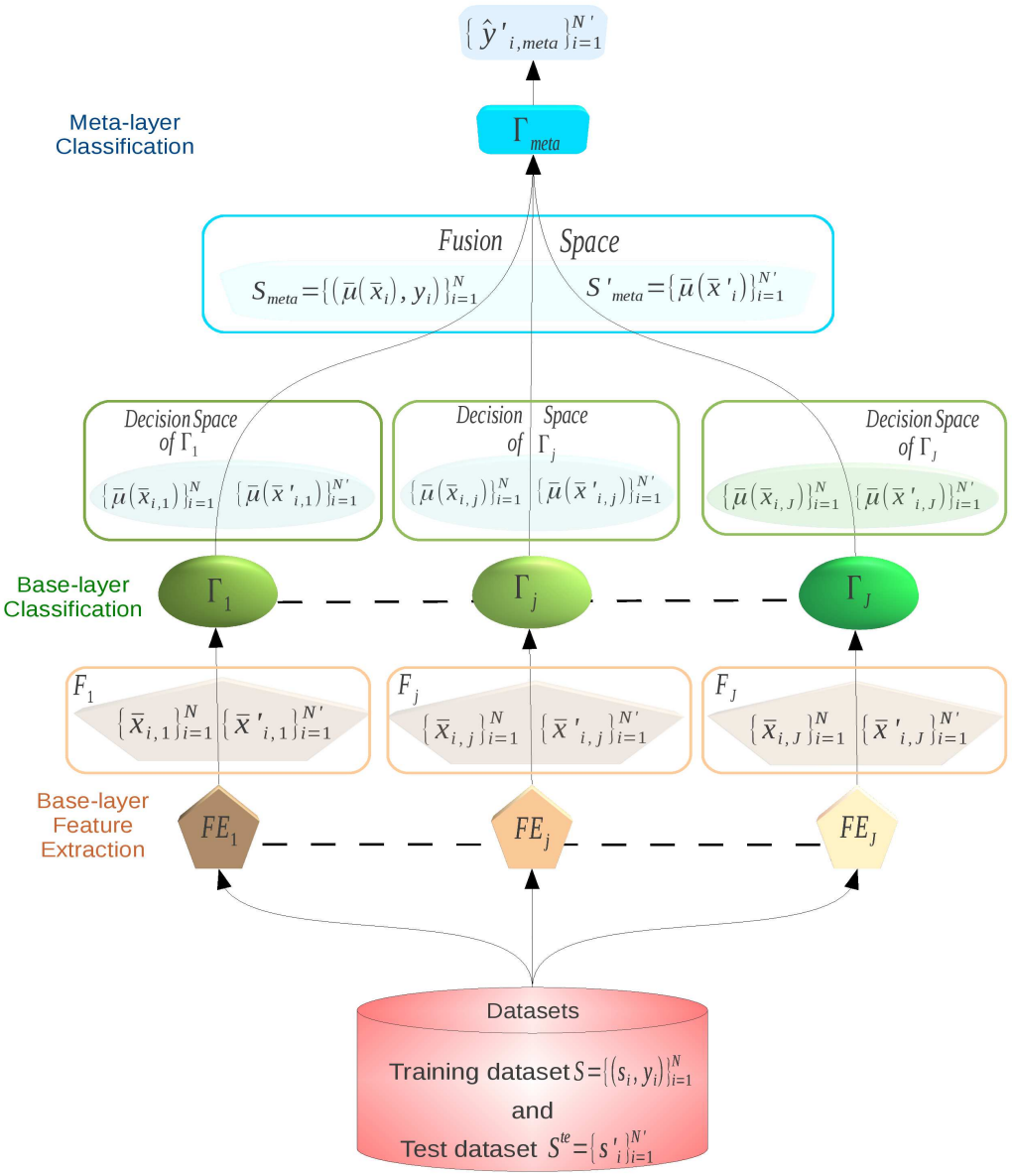


Fig. 1: Flowchart of the FSG architecture (see text for details).

base-layer classifier employed on F_j as

$$\hat{y}_{i,j} = \arg\max_{\omega_c}(\bar{\mu}(\bar{x}_{i,j})),$$

where $\bar{\mu}(\bar{x}_{i,j}) = [\mu_c(\bar{x}_{i,j})]_{c=1}^{C}$. The training performance of the j^{th} base-layer classifier is computed as $Perf_j^{tr} = \frac{1}{N}\sum_{i=1}^{N}\delta_{\hat{y}_{i,j}}(S_j)$, where $\delta_{\hat{y}_{i,j}}(S_j)$ is the Kronecker delta which takes the value 1 when the j^{th} base-layer classifier correctly classifies a sample $s_i \in S$ such that $y_i \equiv \hat{y}_{i,j}$.

In the test step, class membership value $\mu_c(\bar{x}'_{i,j})$ of each test sample s'_i obtained from the test set $S^{te} = \{s'_i\}_{i=1}^{N'}$ is computed using (8) with a set of test feature vectors $S_j^{te} = \{\bar{x}'_{i,j}\}_{i=1}^{N'}$ and S_j in each classifier $\Gamma_j, \forall j = 1, 2, \ldots, J$ (see Fig. 1). Note that the posterior probabilities are approximated by

$$P(\omega_c|\bar{x}'_{i,j}) \approx \mu_c(\bar{x}'_{i,j}). \tag{10}$$

If a set of labeled test samples $\{y'_i\}_{i=1}^{N'}$ is available, then the test performance is computed as $Perf_j^{te} = \frac{1}{N'} \sum_{i=1}^{N'} \delta_{\hat{y}'_{i,j}}(S_j^{te})$.

The output space of each base-layer classifier is spanned by the class membership vectors $\bar{\mu}(\bar{x}_{i,j}) = [\mu_c(\bar{x}_{i,j})]_{c=1}^{C}$ and $\bar{\mu}(\bar{x}'_{i,j}) = \left[\mu_c(\bar{x}'_{i,j})\right]_{c=1}^{C}$ of each sample $s_i \in S$ and $s'_i \in S^{te}$ (see Fig. 1). It should be noted that the class membership vectors satisfy

$$\sum_{c=1}^{C} \mu_c(\bar{x}_{i,j}) = 1 \;\text{ and }\; \sum_{c=1}^{C} \mu_c(\bar{x}'_{i,j}) = 1, \;\; \forall s \in S, s' \in S^{te}, j = 1, 2, \ldots, J.$$

This equation aligns each sample on the surface of a simplex at the output space of a base-layer classifier, which is called the *decision space* of that classifier. Therefore, a base-layer classifier can be considered as a mapping from the input feature space of any dimension into a point on a simplex in a C (number of classes) dimensional decision space. For $C = 2$, the simplex is reduced to a line.

Meta-Layer: When the posteriori probabilities are approximated by the fuzzy class membership values, (7) can be approximated as follows;

$$d(s'_i, s_i) \approx \sum_{j=1}^{J} \|\bar{\mu}(\bar{x}_{i,j}) - \bar{\mu}(\bar{x}'_{i,j})\|_2^2. \tag{11}$$

In order to minimize $d(s'_i, s_i)$, the class-membership vectors $\bar{\mu}(\bar{x}_{i,j})$ and $\bar{\mu}(\bar{x}'_{i,j})$ obtained at the output of each base-layer classifier Γ_j, are concatenated to construct $\bar{\mu}(\bar{x}_i) = [\bar{\mu}(\bar{x}_{i,j})]_{j=1}^{J}$ and $\bar{\mu}(\bar{x}'_i) = \left[\bar{\mu}(\bar{x}'_{i,j})\right]_{j=1}^{J}$, for all training and test samples in a feature space called *fusion space* for a meta-layer classifier Γ_{meta}. The fusion space consists of CJ dimensional feature vectors $\bar{\mu}(\bar{x}_i)$ and $\bar{\mu}(\bar{x}'_i)$ which form the training dataset $S_{meta} = \{(\bar{\mu}(\bar{x}_i), y_i)\}_{i=1}^{N}$ and the test dataset $S'_{meta} = \{\bar{\mu}(\bar{x}'_i)\}_{i=1}^{N'}$ for the meta-layer classifier Γ_{meta} as shown in Fig. 1. Note that $\sum_{j=1}^{J} \sum_{c=1}^{C} \mu_c(\bar{x}_{i,j}) = J$ and $\sum_{j=1}^{J} \sum_{c=1}^{C} \mu_c(\bar{x}'_{i,j}) = J$.

Finally, at the *meta-layer* of the suggested FSG, a fuzzy k-NN classifier Γ_{meta} labels an unknown sample by minimizing the distance

$$d(s'_i, s_i) \approx \|\bar{\mu}(\bar{x}_i) - \bar{\mu}(\bar{x}'_i)\|_2^2 \tag{12}$$

using (8). Note that, if $F = F_j$ for $j \in \{1, 2, \ldots, J\}$, then (12) is reduced to (4). Meta-layer performances are computed using $Perf^{tr}_{meta} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{y}_{i,meta}}(S_{meta})$ and $Perf^{te}_{meta} = \frac{1}{N'}\sum_{i=1}^{N'} \delta_{\hat{y}'_{i,meta}}(S'_{meta})$. An algorithmic description of the FSG is given in Algorithm 1.

input : Training set $S = \{(s_i, y_i)\}_{i=1}^{N}$, test set $S^{te} = \{s'_i\}_{i=1}^{N'}$ and J feature extractors FE_j, $\forall j = 1, 2, \ldots, J$.
output: Predicted class labels of the test samples $\{\hat{y}'_{i,meta}\}_{i=1}^{N'}$.
foreach $j = 1, 2, \ldots, J$ **do**
1 Extract features $\{\bar{x}_{i,j}\}_{i=1}^{N}$ and $\{\bar{x}'_{i,j}\}_{i=1}^{N'}$ using FE_j;
2 Compute $\{\bar{\mu}(\bar{x}_{i,j})\}_{i=1}^{N}$ and $\{\bar{\mu}(\bar{x}'_{i,j})\}_{i=1}^{N'}$ in a base-layer fuzzy k-NN classifier Γ_j using (8);
end
3 Construct $S_{meta} := \{(\bar{\mu}(\bar{x}_i), y_i)\}_{i=1}^{N}$ and $S'_{meta} := \{\bar{\mu}(\bar{x}'_i)\}_{i=1}^{N'}$;
4 Employ meta-layer classification using S_{meta} and S'_{meta} to predict $\{\hat{y}'_{i,meta}\}_{i=1}^{N'}$;

Algorithm 1: Fuzzy Stacked Generalization.

As it is stated in the previous section, minimization of (12) enables us to minimize the expected difference between the large sample and N-sample errors in a fusion space $F = F_1 \times F_2 \times \ldots \times F_J$. Therefore, the proposed FSG reformulates the decision fusion problem as the distance learning problem suggested by Short and Fukunaga [23].

VI. Computational Complexity of FSG

In the analysis of the computational complexity of the proposed FSG algorithm, computational complexities of feature extraction algorithms are ignored assuming that the feature sets are already computed and given.

The computational complexity of the Fuzzy Stacked Generalization algorithm is dominated by the number of samples N. The computational complexity of a base-layer classifier is $O(ND_j)$, $\forall j = 1, 2, \ldots, J$. If each base-layer classifier is implemented by an individual processor in parallel, then the computational complexity of base-layer classification process is $O(N\tilde{D})$, where $\tilde{D} = \max\{D_j\}_{j=1}^{J}$. In addition, the computational complexity of a meta-layer

classifier which employs a fuzzy k-nn is $O(NJC)$. Therefore, the computational complexity of the FSG is $O(\max\{N\tilde{D}, NJC\})$.

VII. Experimental Analysis

In this section, three sets of experiments are performed to analyze the behavior of the suggested FSG and to compare its performance with the state of the art ensemble learning algorithms.

1) The first set of experiments is performed on the artificial dataset in order to analyze the relationships between the performance of the base-layer classifiers and the overall FSG in a controlled environment where the collaboration among the base-layer classifiers is measured by a *shearability metric*. Then, we examine the geometric properties of the transformations from base-layer input feature spaces to base-layer output decision spaces and fusion space.
2) Next, benchmark pattern classification datasets such as Breast Cancer, Diabetis, Flare Solar, Thyroid, German, Titanic [24], [25], [26], [27], [38], [39], Caltech 101 Image Dataset [40] and Corel Dataset [13] are used to compare the classification performances of the proposed approach and state of the art supervised ensemble learning algorithms. We use the same data splitting of the benchmark datasets suggested in [24], [25] to enable the reader to compare our results with the aforementioned distance learning methods referring to [24], [25].
3) Finally, we examine FSG in a real-world target detection problem using a multi-modal dataset, collected by a video camera and microphone in an indoor environment to detect two moving targets. The problem is defined as a four-class classification problem, where each class represents absence or presence of the targets in the environment. In addition, we analyze the statistical properties of the feature spaces at the base-layer and meta-layer by comparing the first order entropies of the distributions of the feature vectors.

In the experiments, fuzzy k-NN algorithm is implemented both in Matlab[1] and C++. For C++ implementations, a fuzzified modification of a GPU-based parallel k-NN is used [41].

[1]A sample Matlab implementation is available on https://github.com/meteozay/fsg.git

k values of the fuzzy k-NN classifiers are optimized by searching $k \in \{1, 2, \ldots, \sqrt{N}\}$ using cross validation, where N is the number of samples in a training dataset. Classification performance of the FSG is compared with that of the state of the art ensemble learning algorithms, such as Adaboost [8], Random Subspace [9] and Rotation Forest [10]. Weighted majority voting is used as the combination rule in Adaboost. Decision trees are implemented as the weak classifiers in both Adaboost and Rotation Forest, and k-NN classifier is implemented as the weak classifier in Random Subspace. The number of weak classifiers $Num_{weak} \in \{1, 2, \ldots, 2D\}$ is selected using cross-validation in the training set, where $D = \sum_{j=1}^{J} D_j$ is the dimension of the feature space of the aggregated feature vectors of the samples in the datasets. Adaboost and Random Subspace algorithms are implemented using Statistics Toolbox of Matlab.

A. Experiments on Artificial Datasets

Nearest neighbor algorithms have been studied by many researchers. In [35], Cover and Hart used an elegant example, which is revised by Devroye, Gyorfi and Lugosi [42]. Later, Hastie and Tibshirani [43] used the results of the example in order to define a metric to minimize the difference between the N-sample and large-sample errors. Since the minimization of error difference is one of the motivations of FSG, a similar experimental setup is designed in order to analyze the performance of FSG.

In the example, feature vectors of the samples of a training dataset $\{(s_i, y_i)\}_{i=1}^{N}$ are grouped in two disks with centers $\bar{o}_1$ and $\bar{o}_2$, which represent the class groups ω_1 and ω_2 such that $\| \bar{o}_1 - \bar{o}_2 \|_2 \geq \sigma_{BC}^{1,2}$ in a two dimensional feature space, where $\sigma_{BC}^{1,2}$ is the between-class variance.

The feature vectors of the samples in the datasets are generated using a circular Gaussian distribution with fixed radius in $D_j = 2$ dimensional feature spaces F_j, $j = 1, 2, \ldots, J$. While constructing the datasets, $\sigma_{BC}^{1,2}$ is varied in a systematical way in order to observe the effect of the class overlaps on the classification performance. This task is achieved by fixing the covariance matrix Σ_c for all the classes, and changing the mean values of the distributions of individual classes, which varies the between-class variances $\sigma_{BC}^{c,c'}$, $\forall c \neq c'$, $c = 1, 2, \ldots, C$, $c' = 1, 2, \ldots, C$.

1) Sample Shareability Property and Shareability Measure: In order to measure the degree of cooperation among the base-layer classifiers, we introduce a measure, called *shareability*. A sample set is called *shareable* by the base-layer classifiers if each sample in the dataset can be classified correctly by at least one of the base-layer classifiers. Experimental evidence indicates that when the dataset is shareable, the base-layer classifiers cooperate to boost the performance of the FSG. We also observe that the performance of FSG decreases as the number of samples which are correctly classified by at least one base-layer classifier is increased, in other words, shareability is decreased. The *degree of shareability* is measured by $\hat{Ave}_{corr}$ which is the average number of samples that are correctly classified by at least one base-layer classifier in a dataset.

input : The number of feature spaces J, the number of classes C, the mean value vectors $\bar{o}_c$ and the within class variances Σ_c of the class conditional distributions, $\forall c = 1, 2, \ldots, C$.
output: Training dataset S_j, and test dataset $S_j^{te} \cup \{y_i'\}_{i=1}^{N'}$, $\forall j = 1, 2, \ldots, J$.
foreach $j = 1, 2, \ldots, J$ **do**
  foreach $c' = 1, 2, \ldots, C$ **do**
1 Initialize $\hat{o}_{c'}$;
    foreach $c = 1, 2, \ldots, C$ **do**
      repeat
2 Generate feature vectors using a circular Gaussian distribution;
3 $\sigma_{BC}^{c,c'} \leftarrow \|\bar{o}_c - \hat{o}_{c'}\|$;
4 $\hat{o}_{c'} \leftarrow \bar{o}_c + \frac{1}{10}\sigma_{BC}^{c,c'}$;
      until $\sigma_{BC}^{c,c'} = 0$;
    end
  end
end
5 Randomly split the feature vectors into two datasets, namely test and training datasets.

Algorithm 2: Artificial dataset generation algorithm.

In order to observe the rate of performance boost of the proposed FSG as a function of the shareability measure, initially, feature spaces are generated to construct classifiers which are *expert* on a specific class with shareability measure $\hat{Ave}_{corr} = 1$. In other words, each classifier is dedicated to correctly classify one of the categories. Then, the shareability measure is gradually decreased.

The dataset generation method is given in Algorithm 2. The feature vectors of the samples belonging to different classes are first generated apart from each other to assure the linear

separability in the initialization step. Then the distances between the mean values of the distributions are gradually decreased. The ratio of decrease is selected as one tenth of between-class variance of distributions for each class pair ω_c and $\omega_{c'}$, $\forall c \neq c'$, $c = 1, 2, \ldots, C$, $c' = 1, 2, \ldots, C$, which is $\frac{1}{10}\sigma_{BC}^{c,c'}$, where $\sigma_{BC}^{c,c'} = \|\bar{o}_c - \bar{o}_{c'}\|$. At each epoch, only the mean value of the distribution of one of the classes approaches to the mean value of that of another class, while keeping the mean values of the distributions of the rest of the classes fixed.

2) Performance Analysis on Artificial Datasets: In this set of the experiments, 7 base-layer classifiers are used to classify samples belonging to 12 categories. The number of samples belonging to each class ω_c is taken as 250. 2-dimensional feature spaces are fed to each base-layer classifier as input with $250 \times 12 = 3000$ samples. Feature sets S_j and S_j^{te} are prepared with fixed and equal values of the covariance matrices Σ_c of the class conditional distributions in F_j, $\forall j = 1, 2, \ldots, 7$, as

$$\Sigma_c = \begin{pmatrix} 5 & 5 \\ 5 & 5 \end{pmatrix}, \forall c = 1, 2, \ldots, 12.$$

In Tables I,II, III and IV, the performances of individual classifiers and the proposed FSG algorithm are given for the shareability measures $\hat{Ave}_{corr} = 1$, $\hat{Ave}_{corr} = 0.9$, $\hat{Ave}_{corr} = 0.8$, $\hat{Ave}_{corr} = 0.7$, respectively, on the datasets generated by Algorithm 2. Recall that for $\hat{Ave}_{corr} = 1$, the datasets are constructed in such a way that each sample is correctly recognized by at least one of the base-layer classifiers. Note that, in Table I, although the classification performances of individual classifiers are in between $53\% - 66\%$, the performance of the FSG reaches to 99.9%. In Tables II, III and IV, we observe that the performances decrease as the shareability measure $\hat{Ave}_{corr}$ decreases. This behavior of the FSG is geometrically analyzed in the experiments of the next subsection.

3) Geometric Analysis of Feature, Decision and Fusion Spaces on Artificial Datasets: Recall that the membership values of the samples lie on the surface of a simplex in the C-dimensional decision space of each base-layer classifier. In practice, the highest membership value of a feature (membership) vector $\bar{\mu}(\bar{x}_j)$ represents the *predicted* class label $\hat{y}_j$ of a sample s in F_j, $\forall j = 1, \ldots, J$, and the membership vector of a correctly classified sample is expected to accumulate around the *correct* (and *target*) vertex of the simplex, which represents

TABLE I: Comparison of the classification performances (%) of the base-layer classifiers with respect to the classes (***Class-ClassID***) and the performances of the FSG, when the shareability is $\hat{Ave}_{corr} = 1$.

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	FSG
Class-1	66.0%	63.6%	67.6%	62.8%	61.6%	85.6%	50.0%	**100.0**%
Class-2	67.2%	60.8%	49.6%	50.8%	***98.4***%	38.4%	36.8%	**100.0**%
Class-3	54.4%	58.8%	50.8%	***85.2***%	72.4%	53.6%	47.6%	**99.2**%
Class-4	66.8%	64.0%	***96.8***%	66.4%	61.6%	22.8%	37.6%	**100.0**%
Class-5	60.8%	***90.0***%	56.0%	63.6%	75.2%	38.8%	48.4%	**100.0**%
Class-6	***91.6***%	57.2%	69.6%	54.0%	66.0%	43.6%	73.6%	**100.0**%
Class-7	57.2%	55.2%	65.2%	57.6%	60.8%	37.2%	***94.4***%	**100.0**%
Class-8	78.4%	75.6%	86.0%	69.2%	54.4%	61.6%	***97.6***%	**100.0**%
Class-9	40.8%	41.2%	36.0%	36.0%	32.8%	26.0%	***99.6***%	**100.0**%
Class-10	44.0%	32.4%	32.0%	38.0%	37.6%	43.2%	***95.6***%	**100.0**%
Class-11	32.0%	35.2%	33.6%	40.0%	39.6%	***92.8***%	38.8%	**99.6**%
Class-12	37.6%	39.6%	34.4%	52.0%	44.4%	***97.2***%	63.6%	**99.6**%
Ave. Perf. (%)	58.0%	56.1%	56.5%	56.3%	58.7%	53.4%	65.3%	**99.9**%

TABLE II: Comparison of the classification performances (%) of the base-layer classifiers with respect to the classes (***Class-ClassID***) and the performances of the FSG, when shareability is $\hat{Ave}_{corr} = 0.9$.

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	FSG
Class-1	***97.2***%	67.6%	68.4%	69.6%	28.0%	53.6%	65.6%	**100.0**%
Class-2	***96.8***%	63.2%	63.6%	41.6%	67.6%	44.4%	30.0%	**100.0**%
Class-3	56.4%	***95.2***%	57.2%	66.8%	56.8%	47.2%	66.4%	**99.6**%
Class-4	60.8%	***98.0***%	22.8%	30.8%	62.0%	24.4%	46.0%	**100.0**%
Class-5	56.8%	24.0%	***96.8***%	27.2%	44.8%	38.8%	50.4%	**100.0**%
Class-6	32.8%	68.4%	***97.6***%	71.2%	57.2%	43.6%	14.0%	**100.0**%
Class-7	54.0%	65.6%	74.4%	***96.8***%	52.4%	36.8%	24.4%	**99.6**%
Class-8	77.2%	43.6%	29.6%	***98.4***%	48.0%	65.6%	27.6%	**99.6**%
Class-9	45.2%	34.0%	35.2%	35.2%	***98.8***%	24.8%	29.2%	**100.0**%
Class-10	40.0%	33.6%	22.4%	47.6%	***90.4***%	33.6%	18.0%	**100.0**%
Class-11	49.2%	28.4%	38.0%	28.0%	38.4%	***100.0***%	26.0%	**100.0**%
Class-12	34.8%	34.4%	22.4%	34.4%	44.4%	65.2%	***98.8***%	**100.0**%
Ave. Perf. (%)	58.4%	54.6%	52.3%	53.9%	57.4%	48.1%	41.3%	**99.9**%

the *target* class label of that sample. Concatenation operation, used to form a CJ-dimensional fusion space at the input of the meta-layer classifier creates a CJ-dimensional simplex. The membership values of the correctly classified samples, this time, form even a more compact cluster around each vertex of the simplex, whereas misclassified samples are scattered all over the surface. This fact is geometrically depicted in the following example.

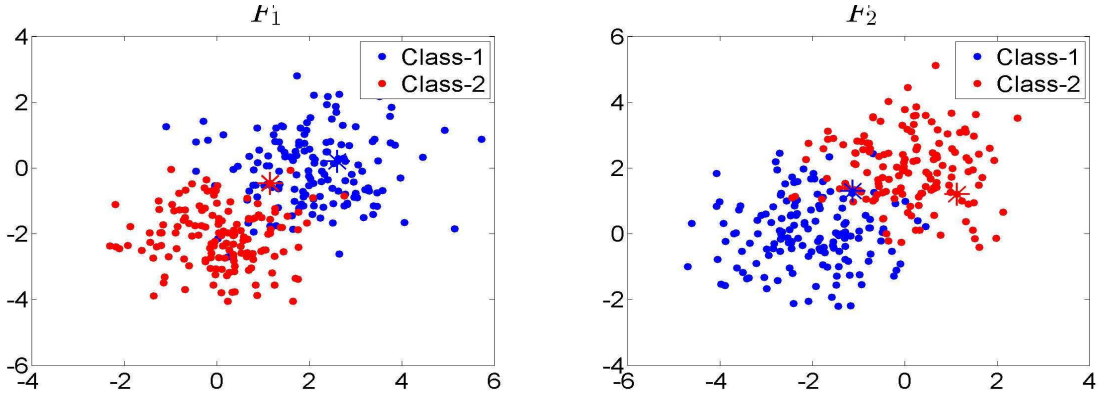
TABLE III: Comparison of the classification performances (%) of the base-layer classifiers with respect to the classes (***Class-ClassID***) and the performances of the FSG, when shareability is $\hat{Ave}_{corr} = 0.8$.

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	FSG
Class-1	***82.8***%	63.6%	66.0%	71.2%	32.0%	54.0%	67.2%	**99.6**%
Class-2	***73.2***%	63.6%	48.0%	34.4%	51.6%	37.6%	29.6%	**97.2**%
Class-3	55.2%	***78.0***%	59.6%	51.2%	62.4%	46.8%	69.6%	**98.4**%
Class-4	61.2%	***82.0***%	26.0%	31.2%	44.4%	17.6%	52.8%	**98.4**%
Class-5	53.2%	23.2%	***76.8***%	29.6%	41.2%	39.6%	45.2%	**100.0**%
Class-6	24.8%	66.4%	***87.2***%	62.0%	56.4%	42.4%	21.2%	**98.8**%
Class-7	54.0%	63.2%	54.8%	***88.4***%	55.2%	36.8%	23.6%	**98.4**%
Class-8	***80.8***%	39.2%	22.8%	74.8%	45.2%	63.2%	23.6%	**96.4**%
Class-9	39.6%	33.2%	33.2%	29.6%	***83.6***%	21.6%	29.6%	**99.2**%
Class-10	38.4%	35.6%	30.8%	47.6%	***82.8***%	38.0%	24.0%	**99.2**%
Class-11	33.2%	30.0%	30.8%	30.4%	38.8%	***84.4***%	29.6%	**96.4**%
Class-12	40.4%	33.2%	28.0%	40.4%	32.4%	58.8%	***81.2***%	**99.2**%
Ave. Perf. (%)	53.1%	50.9%	47.0%	49.2%	52.2%	45.1%	41.4%	**98.4**%

TABLE IV: Comparison of the classification performances (%) of the base-layer classifiers with respect to the classes (***Class-ClassID***) and the performances of the FSG, when shareability is $\hat{Ave}_{corr} = 0.7$.

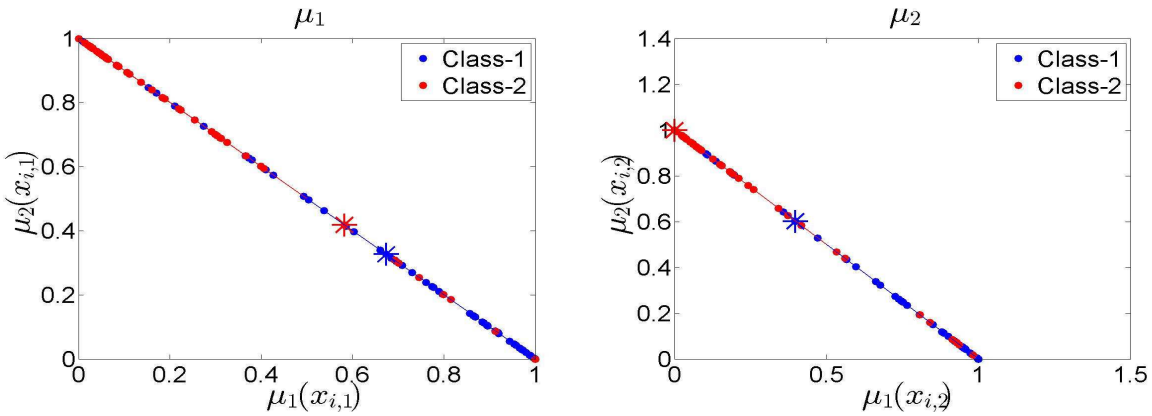
	F_1	F_2	F_3	F_4	F_5	F_6	F_7	FSG
Class-1	***75***%	42%	68%	52%	36%	62%	46%	**99**%
Class-2	***64***%	45%	41%	38%	43%	37%	32%	**98**%
Class-3	46%	***72***%	60%	40%	39%	52%	46%	**88**%
Class-4	68%	***72***%	23%	33%	45%	17%	59%	**98**%
Class-5	54%	22%	***70***%	28%	40%	42%	32%	**100**%
Class-6	22%	68%	***74***%	50%	46%	28%	18%	**97**%
Class-7	65%	62%	50%	***72***%	44%	34%	20%	**96**%
Class-8	55%	30%	25%	***75***%	44%	61%	18%	**89**%
Class-9	36%	24%	36%	30%	***67***%	32%	23%	**100**%
Class-10	42%	32%	24%	27%	***74***%	32%	21%	**98**%
Class-11	31%	17%	34%	16%	38%	***70***%	26%	**95**%
Class-12	33%	28%	27%	41%	38%	67%	***68***%	**100**%
Ave. Perf. (%)	49.3%	42.9%	44.3%	41.8%	46.1%	44.4%	34.2%	**96.4**%

Consider an artificial dataset consisting of $C = 2$ classes each of which consists of 250 samples represented in $J = 2$ distinct feature spaces. In the base-layer feature spaces shown in Fig. 2, the classes have Gaussian distribution with substantial overlaps where the mean



(a) F_1 (b) F_2

Fig. 2: Feature vectors in (a) F_1 and (b) F_2. Features of two randomly selected samples are indicated by $(*)$ to follow them at the decision spaces of base-layer classifiers and the fusion space of meta-layer classifier.



(a) (b)

Fig. 3: Membership vectors obtained at the decision spaces of base-layer classifiers: (a) the first classifier Γ_1 and (b) the second classifier Γ_2. The locations of the features of randomly selected samples of Fig. 2 are indicated by $(*)$, at each simplex.

values and covariance matrices are

$$\Omega_1 = \begin{pmatrix} 2 & 0 \\ 0 & -2 \end{pmatrix}, \ \Sigma_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \text{ and } \Omega_2 = \begin{pmatrix} -2 & 0 \\ 2 & 2 \end{pmatrix}, \ \Sigma_2 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

for the first and the second feature spaces, respectively. The features of the samples belonging to the first and the second class are represented by blue and red dots, respectively. Features of two randomly selected samples, which are misclassified by one of the base-layer classifiers and correctly classified by the meta-layer classifier, are shown by star $(*)$ markers. In the

feature spaces, each sample is correctly classified by at least one base-layer fuzzy k-NN classifier with $k = 3$. The classification performances of the base-layer classifiers are 91% and 92%, respectively. The classification performance of the FSG is 96%.

The membership values lie on a line in the decision spaces of two base-layer classifiers, as depicted in Fig. 3. In these figures, the decisions of the classifiers are also depicted for individual samples. For instance, the sample marked with red star, s_1, is misclassified by the first classifier as shown in Fig. 3.a, but correctly classified by the second classifier as shown in Fig. 3.b. In addition, the feature of the sample marked with blue star, s_2, is correctly classified by the first classifier as shown in Fig. 3.a, but misclassified by the second classifier as shown in Fig. 3.b.

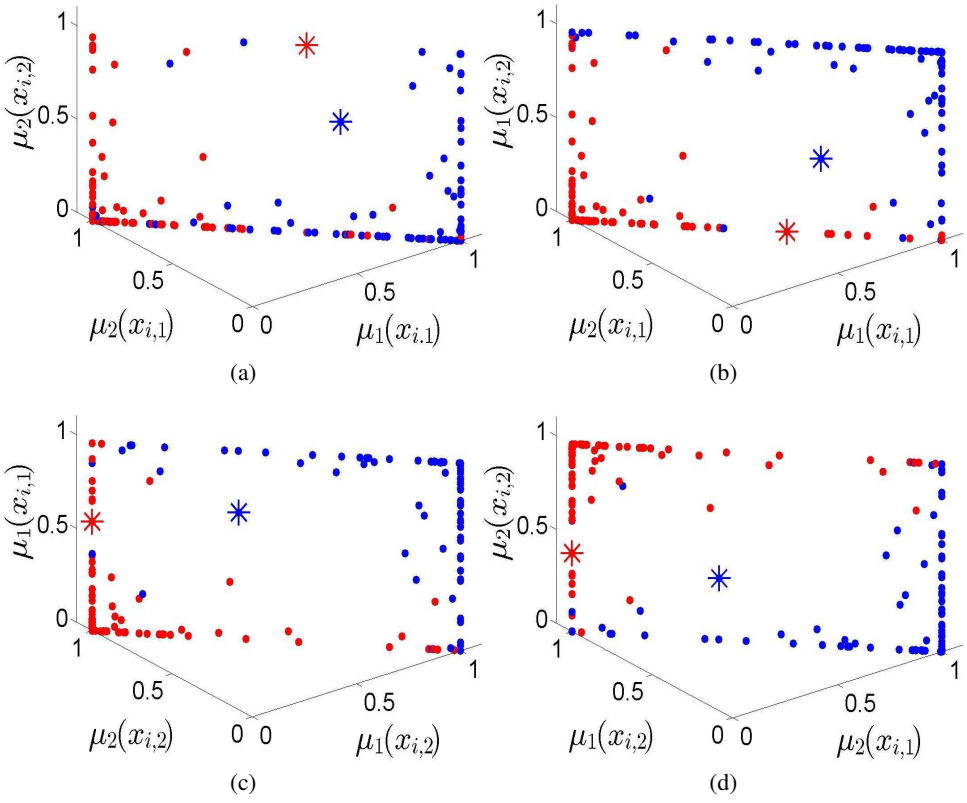


Fig. 4: The relationships among (a) $\mu_1(x_i, 1)$, $\mu_2(x_i, 1)$, $\mu_2(x_i, 2)$, (b) $\mu_1(x_i, 1)$, $\mu_2(x_i, 1)$, $\mu_1(x_i, 2)$, (c) $\mu_1(x_i, 2), \mu_(x_i, 2), \mu_1(x_i, 1)$, and (d) $\mu_2(x_i, 1)$, $\mu_1(x_,2)$, $\mu_2(x_i, 2)$, are visualized. The locations of the features of randomly selected samples of Fig. 2 are indicated by $(*)$ in the subspaces of the fusion space.

A 4 (2×2) dimensional fusion space is created at the meta-layer. In order to visualize the distribution of 4-dimensional membership vectors of samples in the fusion space, four

different subspaces, each of which is a 3-dimensional Euclidean space, are selected. Fig. 4 displays different combinations of the subspaces and the membership vectors obtained from each classifier. Notice that the concatenation operation forms planes in these subspaces accumulating the correctly classified samples around the edges and the vertices. Therefore, features of the samples which are correctly classified by at least one base-layer classifier are located closer to one of the *correct* vertices or edges in the fusion space. This fact is depicted in Fig. 4, where the feature of the sample indicated by red star is located closer to the edges of the second class in Fig. 4.b, c, d. On the other hand, the feature of the sample indicated by blue star is located closer to the edges of the first class in Fig. 4.a, c, d. Both of these samples are correctly labeled by the meta-layer fuzzy k-NN classifier.

B. Experiments on Benchmark Datasets

In the experiments, classification performances of $k = 1$ nearest neighbor rule, Fuzzy Stacked Generalization (FSG), and the state of the art algorithms, Adaboost, Random Subspace (RS) and Rotation Forest (RF), are compared using benchmark datasets.

Experiments on the benchmark datasets are performed in two groups:

1) **Multi-attribute Datasets**: Feature vectors consisting of multiple attributes reside in a single feature space $F_j = F_{1j} \times \ldots \times F_{aj} \times \ldots \times F_{Aj}$, where A is the number of attributes. In these experiments, FSG is implemented by employing individual base-layer classifiers on a feature space F_{aj} consisting of an individual attribute. Therefore, the dimension of the feature vectors in the fusion space of the FSG is CA.
2) **Multi-feature Datasets**: Each base-layer classifier of FSG is employed on an individual feature space F_j, $\forall j = 1, 2, \ldots, J$. Therefore, the dimension of the feature vectors in the fusion space of the FSG is CJ.

State of the art algorithms are employed on an aggregated feature space $F = F_1 \times F_2 \times \ldots \times F_J$ which contains feature vectors with dimension A and $D = \sum_{j=1}^{J} D_j$ in multi-attribute and multi-feature experiments, respectively.

1) Experiments on Multi-attribute Datasets: In the experiments, Breast Cancer (BCancer), Diabetis, Flare Solar (FSolar), Thyroid, German, Titanic [24], [26], [27], [38], [39] datasets are used as multi-attribute datasets. The numbers of attributes of the feature vectors of the

TABLE V: Classification performances of the algorithms on Multi-attribute Datasets.

Datasets	**Titanic**	**Thyroid**	**Diabetis**	**FSolar**	**BCancer**	**German**
Num. of Att.(A)	3	5	8	9	9	20
Adaboost	75.06%	93.10%	75.98%	66.21%	74.87%	75.89%
Rotation Forest	70.14%	95.64%	72.43%	62.75%	70.58%	74.81%
Random Subspace	74.83%	94.78%	74.40%	65.04%	74.08%	75.17%
1 NN	75.54%	95.64%	69.88%	60.58%	67.30%	71.12%
FSG	76.01%	96.41%	77.42%	67.33%	75.51%	75.30%

samples in the datasets are given in Table V. Training and test datasets are randomly selected from the datasets using the data splitting scheme of [24], [25]. The experiments are repeated 100 times, and the average performance values are given in Table V.

An interesting observation on Table V is that the $k = 1$ nearest neighbor rule outperforms various well-known ensemble learning algorithms such as Adaboost and Rotation Forest, if the number of attributes is small, e.g. $A = 3$. The classification performance of the nearest neighbor rule decreases as A increases due to the curse of dimensionality problem of the nearest neighbor algorithms [34]. Since the dimension of the feature vectors in the fusion space is CA, the dimensionality curse can be observed in the fusion space of the FSG as A increases. We further analyze the relationship between classification performances, the number of classes and classifiers in the next subsection.

2) Experiments on Multi-feature Datasets: In this section, the algorithms have been analyzed on Corel Dataset[2] consisting of 599 classes and Caltech 101 Dataset consisting of 102 classes.

7.2.2.1 Experiments on Corel Dataset

Corel Dataset experiments are performed by randomly selecting samples belonging to 10 to 30 classes (out of 599 classes) each of which contains $97 - 100$ samples from the dataset. 50 of the samples belonging to each class are used for training, and the remaining samples are used for testing. 4 to 8 feature combinations of Haar and 7 of MPEG-7 visual features [44], [45] are used. The feature set combinations are selected as follows:

[2]The dataset is available on https://github.com/meteozay/Corel_Dataset.git

- **4 Features (4FS)**: Color Structure, Color Layout, Edge Histogram, Region-based Shape,
- **5 Features (5FS)**: **4 Features (4FS)** and Haar,
- **6 Features (6FS)**: **5 Features (5FS)** and Dominant Color,
- **7 Features (7FS)**: **6 Features (6FS)** and Scalable Color,
- **8 Features (8FS)**: **7 Features (7FS)** and Homogenous Texture.

The selected MPEG-7 features have high variance and a well-balanced cluster structure [44]. In addition, the feature vectors in the descriptors satisfy i.i.d. (independent and identically distributed) conditions and provide high between class variances [44]. Therefore, the statistical properties of the feature spaces provide wealthy information variability.

Experiments are performed in two groups. In the first group, the samples are randomly selected from the following pre-defined classes:

- **10 Class Classification**: New Guinea, Beach, Rome, Bus, Dinosaurs, Elephant, Roses, Horses, Mountain, and Dining,
- **15 Class Classification**: Classes used in **10 Class Classification** together with Autumn, Bhutan, California Sea, Canada Sea and Canada West,
- **20 Class Classification**: Classes used in **15 Class Classification** together with China, Croatia, Death Valley, Dogs and England.

TABLE VI: Classification performances (%) of the algorithms on the Corel Dataset with varying number of features and classes.

	Algorithms	**4FS**	**5FS**	**6FS**	**7FS**	**8FS**
10-Class Experiments	**Adaboost**	63.0%	63.6%	63.2%	66.6%	67.2%
	Rotation Forest	76.2%	74.4%	74.6%	76.6%	78.2%
	Random Subspace	78.1%	77.5%	75.8%	76.9%	75.5%
	FSG	85.6%	86.8%	85.6%	85.8%	85.8%
15-Class Experiments	**Adaboost**	42.2%	45.5%	43.2%	46.8%	46.8%
	Rotation Forest	60.2%	60.6%	60.9%	60.9%	61.3%
	Random Subspace	65.5%	64.1%	59.8%	63.3%	61.8%
	FSG	66.2%	65.3%	62.3%	62.8%	64.5%
20-Class Experiments	**Adaboost**	23.3%	27.0%	27.0%	27.0%	27.0%
	Rotation Forest	47.7%	49.5%	49.5%	49.6%	50.4%
	Random Subspace	48.3%	48.1%	48.1%	48.6%	48.7%
	FSG	52.4%	50.7%	49.9%	50.9%	52.9%

TABLE VII: Classification performances (%) of the algorithms on the Corel Dataset.

C	**Adaboost**	**RF**	**RS**	**1 NN**	**FSG**
	Ave.±Var.	**Ave.±Var.**	**Ave.±Var.**	**Ave.±Var.**	**Ave.±Var.**
2	90.56±9.30%	86.00±0.97%	88.11±0.75%	82.44±2.78%	91.00±0.43%
3	81.33±0.97%	76.27±0.57%	75.87±0.62%	75.27±0.55%	86.97±0.53%
4	73.45±0.54%	69.75±0.81%	70.45±1.27%	69.60±1.10%	83.85±0.59%
5	64.32±0.32%	62.72±0.78%	65.32±0.92%	61.08±0.65%	74.32±0.42%
6	61.17±0.86%	61.67±0.83%	64.20±1.24%	60.50±0.84%	71.90±0.67%
7	54.12±0.67%	58.00±0.51%	62.98±0.45%	56.98±0.55%	68.65±0.44%
8	53.17±0.12%	60.03±0.30%	54.92±2.36%	58.22±0.35%	68.72±0.28%
9	49.02±1.35%	56.98±1.81%	55.89±3.37%	54.98±1.87%	67.82±1.16%
10	39.65±0.65%	48.35±0.27%	47.00±0.35%	47.60±0.58%	59.80±0.37%
12	38.64±0.65%	45.57±0.87%	43.22±1.13%	45.02±0.86%	57.46±0.48%
14	33.16±0.66%	47.16±0.63%	46.81±0.71%	45.76±0.85%	57.87±0.75%
16	29.54±0.17%	40.42±0.24%	41.53±0.29%	39.86±0.31%	52.07±0.44%
18	25.30±0.59%	41.56±0.42%	40.91±0.47%	39.97±0.44%	51.09±0.47%
20	19.46±0.14%	38.27±0.16%	39.98±0.21%	36.25±0.24%	47.77±0.20%
25	16.15±0.23%	35.92±0.42%	35.57±0.63%	33.94±0.37%	45.84±0.42%
30	14.37±0.55%	33.53±0.22%	36.28±0.58%	32.43±0.26%	41.33±0.52%

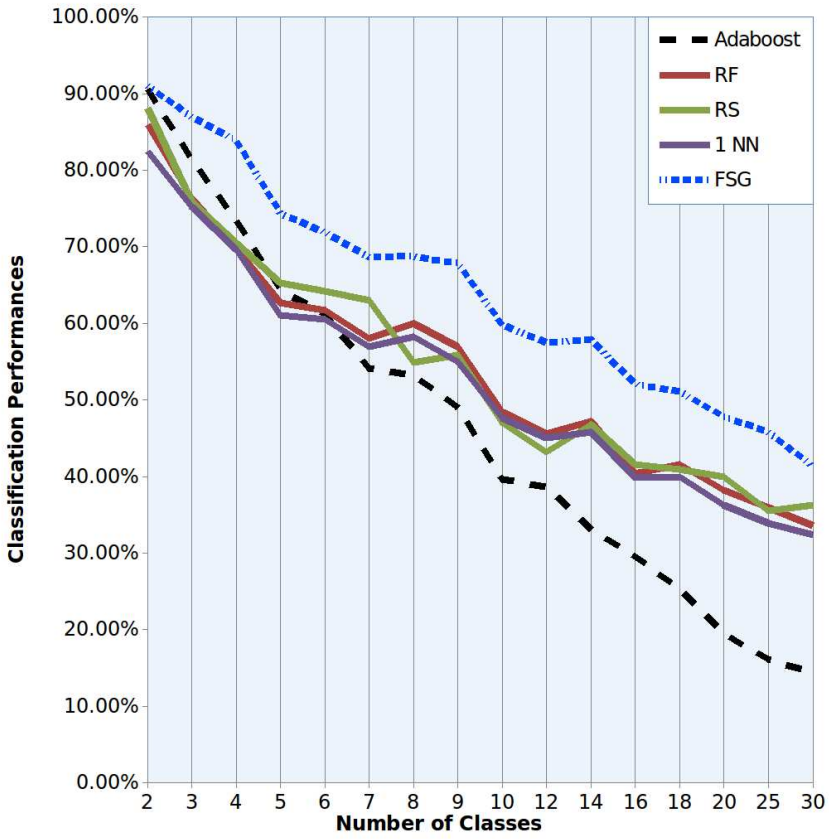


Fig. 5: Classification performances (%) of the algorithms on the Corel Dataset. Note that the best performance is achieved by the FSG algorithm.

The performances of FSG and benchmark algorithms are compared with respect to the selected feature sets in Table VI. Note that the performances of the algorithms which employ majority voting to the classifier decision may decrease as new features are added. For instance, when Dominant Color and Scalable Color features are added to the combination of features in **5FS** to construct **6FS** and **7FS**, the classification performances of the FSG and the Random Subspace, which employ majority voting at the meta-layer classifiers, decrease.

In the second group of experiments, the datasets are constructed by the samples belonging to randomly selected classes. In these experiments, the sample selection procedure is repeated 10 times and the average performance is measured. Average (Ave.) and variance (Var.) of the classification performances of the FSG and benchmark algorithms are given in Table VII. The classification results given in the tables are depicted in Fig. 5.

In the experiments, the performance of the FSG gets better compared to the benchmark algorithms as the number of classes (C) increases. The performance of the Adaboost algorithm decreases faster than the other algorithms as C increases (see Fig. 5). Moreover, the Adaboost algorithm performs better than the other benchmark algorithms for classifying the samples belonging to $C \leq 5$ classes. However, the performances of the Adaboost and the FSG are approximately the same for $C = 2$ class classification. Finally, it is interesting to note that, 1-NN classifier outperforms the Adaboost and is competitive to the other benchmark classifiers for $C \geq 7$.

7.2.2.2 Experiments on Caltech Dataset

In this subsection, the samples belonging to 2 to 10 different classes are randomly selected from the Caltech dataset. The experiments are repeated 10 times for each selection procedure.

In the experiments, the features provided by Gehler and Nowozin [40] are used to construct four feature spaces. Two feature spaces consist of SIFT features extracted on a gray scale and an $\mathcal{HSI}$ image. The third and the fourth feature spaces contain the features extracted using Region Covariance and Local Binary Pattern descriptors. Implementation details of the feature extraction algorithms are given in [40].

The experimental results given in Table VIII show that classification performances of the algorithms do not decrease linearly by increasing number of classes as observed in the experiment with Corel dataset. Note that this non-linear performance variation is observed

for all of the aforementioned algorithms. This behavior may be attributed to the nonlinearity of many interacting parameters of the algorithms.

TABLE VIII: Classification performances of the algorithms on the Caltech Dataset.

C	**Adaboost**	**RF**	**RS**	**1 NN**	**FSG**
	Ave.±Var.	**Ave.±Var.**	**Ave.±Var.**	**Ave.±Var.**	**Ave.±Var.**
2	96.47±0.13%	87.72±2.86%	87.70±1.31%	87.78±2.00%	95.64±0.28%
3	89.68±0.11%	80.90±0.46%	81.20±0.33%	80.90±0.46%	90.46±0.12%
4	81.21±1.55%	74.17±1.82%	76.10±1.73%	72.20±2.62%	85.32±0.70%
5	83.27±0.95%	77.66±0.92%	76.91±1.07%	77.55±1.24%	88.57±0.41%
6	85.14±0.69%	82.73±0.47%	83.42±0.51%	80.97±0.97%	92.15±0.25%
7	77.00±0.55%	76.86±0.32%	76.79±0.49%	76.71±0.25%	88.54±0.23%
8	68.49±1.14%	71.46±0.97%	70.13±1.07%	66.77±2.83%	85.89±0.35%
9	75.48±0.88%	75.90±0.71%	75.93±0.83%	75.69±0.76%	86.28±0.24%
10	64.30±0.34%	65.66±0.20%	65.47±0.18%	62.30±0.30%	81.06±0.23%

C. Experiments for Multi-modal Target Detection

Integration of sensors of multiple modalities by decision fusion algorithms is an important issue for various research fields such as robotics. Decision fusion algorithms, which employ ensemble learning approach such as Adaboost, are only successful in classifying the data sampled from the same distribution. Unfortunately, most of the decision fusion systems may not satisfy this requirement for multi-modal sensor fusion. FSG forms a convenient platform by mapping the data from various modalities into a set of membership values at the base-layer. In this subsection, FSG is implemented for multi-modal target detection problem.

In the experiments, the data acquisition process is accomplished by an audio-visual sensor, which is a webcam with a microphone located in an indoor environment. In this scenario, recordings of the audio and video data are obtained from randomly moving targets T_1 and T_2, i.e. two randomly walking people, in the indoor environment. The problem is defined as the classification of the audio and video frames which represent the presence and absence of two targets moving in the noisy environment, where the other people talking in the environment and the obstacles distributed in the room are the sources of the noise for audio and video data. Four classes, each of which consists of 190 train and 190 test samples, are defined according to the presence and absence of targets T_1 and T_2 in the environment (see:Table

IX). The audio characteristics of the targets are determined with different tunes.

TABLE IX: Classes, which are defined by presence (★) and absence (◯) of targets, T_1 and T_2.

	Class1	Class2	Class3	Class4
T_1	◯	★	◯	★
T_2	◯	◯	★	★

The experiments are designed to achieve complementary expertise of the base-layer classifiers on different classes. For instance, if a target is hidden behind an obstacle such as a curtain (see Fig. 6), then a base-layer classifier which employs audio features can correctly detect the target behind the curtain, even if a base-layer classifier which employs visual features cannot detect the target correctly.

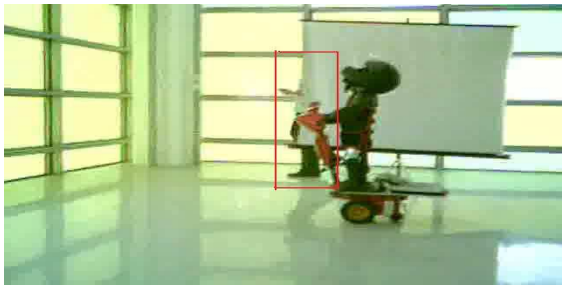


Fig. 6: A sample frame used in the training dataset in which a target (T_1) is hidden behind an obstacle that is a curtain.

In the experiments, two MPEG-7 descriptors, Homogenous Texture (HT) and Color Layout (CL), and three audio descriptors, Fluctuation, Chromagram and Mel-Frequency Cepstral Coefficients (MFCC), [46] are used to extract visual and audio features, respectively [46]. FSG is used for the fusion of the decisions of the classifiers employed on i) **Visual** features using only HT and CL, ii) **Audio** features using only Fluctuation, Chromagram and MFCC, and iii) all ***Audio-Visual*** features.

Experimental results show that the base-layer classifiers employed on visual features perform better than the classifiers employed on audio features for the fourth class. However, the classifiers employed on audio features perform better than the classifiers employed on visual features for the first three classes (see Table X and Table XI). On the other hand, the base-layer classifiers employed on audio features have a better discriminative property

TABLE X: Classification performances for training set.

	Class1	Class2	Class3	Class4	Total
HT	76.84%	67.89%	76.84%	96.30%	79.45%
Color Layout	93.16%	86.84%	84.21%	**97.35%**	90.38%
MFCC	**99.47%**	84.74%	**94.74%**	83.60%	**90.65%**
Chromagram	98.42%	**90.00%**	89.47%	82.01%	89.99%
Fluctuation	94.74%	85.79%	75.79%	52.38%	77.21%
Visual FSG	92.63%	87.37%	84.21%	95.77%	89.99%
Audio FSG	97.89%	93.16%	96.32%	92.59%	94.99%
Audio-Visual FSG	99.47%	97.89%	98.42%	100.00%	98.95%

TABLE XI: Classification performances for test set.

	Class1	Class2	Class3	Class4	Total
HT	54.74%	49.47%	43.75%	**93.12%**	60.91%
Color Layout	76.32%	49.47%	40.63%	83.07%	63.24%
MFCC	92.11%	77.37%	**93.13%**	81.48%	**85,73%**
Chromagram	92.63%	**84.21%**	83.13%	66.67%	81.62%
Fluctuation	**93.68%**	82.63%	75.00%	52.38%	75.99%
Visual FSG	69.47%	54.21%	45.63%	90.48%	65.71%
Audio FSG	90.53%	93.16%	93.13%	79.37%	88.89%
Audio-Visual FSG	93.68%	94.21%	94.37%	97.88%	95.06%

compared to the base-layer classifiers employed on the visual features for the first class. One of the reasons of this observation is that the classifiers employing audio features, which are affected by *audio data noise*, are less sensitive to *feature noise* than the classifiers employed on visual features which are affected by *visual data noise*. In other words, two targets have visual appearance properties similar to the other objects in the environment, and the obstacles (e.g. curtains and doors) block completely the visual appearance of the targets. On the other hand, the targets have different visual appearance properties such that the heights of the targets and colors of their clothes are different from each other. As a result, the base-layer classifiers of the FSG complement each other, and a substantial increase in the classification performance of the FSG is achieved.

Each cell of Table XII and Table XIII represents the number of samples which are misclassified by the fuzzy k-NN classifier for the descriptor given in the i^{th} row, and correctly classified by the classifier for the descriptor given in the j^{th} column, using the training and

TABLE XII: Covariance matrix of the number of correctly classified and misclassified samples in training dataset.

Training Dataset		**Correct Classification**					
		HT	CL	MFCC	Chromagram	Fluctuation	Total
Misclassification	HT	0	137	142	**144**	130	156
	CL	54	0	**64**	59	57	73
	MFCC	57	**62**	0	44	40	71
	Chromagram	**64**	62	49	0	39	76
	Fluctuation	147	**157**	142	136	0	173

TABLE XIII: Covariance matrix of the number of correctly classified and misclassified samples in test dataset.

Test Dataset		**Correct Classification**					
		HT	CL	MFCC	Chromagram	Fluctuation	Total
Misclassification	HT	0	134	247	**249**	233	285
	CL	117	0	**235**	223	216	268
	MFCC	66	**71**	0	52	54	104
	Chromagram	**98**	89	82	0	61	134
	Fluctuation	**123**	**123**	**125**	102	0	175

test datasets, respectively. In the tables, the maximum number of misclassified samples for each descriptor is bolded. For example, 144 samples which are misclassified in HT feature space are correctly classified in Chromagram feature space. The samples that are misclassified in the feature spaces defined by the visual descriptors are correctly classified in the feature spaces defined by the audio descriptors. This is observed when the visual appearances of the targets are degraded by the visual noise, e.g. the targets are completely blocked by an obstacle, such as a curtain, but their sounds are captured by the audio sensor (see Fig. 6).

On the other hand, the samples that are misclassified in the feature spaces defined by the audio descriptors (e.g. Fluctuation and Chromagram) are correctly classified in the feature spaces defined by the visual descriptors (e.g. CL and HT) when there are other objects that generate sounds with audio characteristics similar to the targets in the environment. In this

case, audio features of the targets are affected by audio noise. If the visual sensor can make *clear* measurements on the targets, such that the visual features are not affected by visual noise, then the classifiers employed in the feature spaces defined by the visual descriptors can correctly classify the samples.

1) Statistical Analysis of Feature, Decision and Fusion Spaces on Multi-modal Dataset: In this subsection, class conditional distributions are analyzed in three feature spaces of the proposed FSG, namely, i) in feature spaces at the input of base-layer classifiers, ii) in decision spaces at the output of base-layer classifiers and iii) in fusion space at the input of meta-layer classifier (see Fig. 1). The class conditional distributions are approximated by histograms [47], where the range of the vectors is divided into B intervals, $b = 1, 2, \ldots, B$, with the width w_b of the b^{th} *bin* of a histogram. The probability of a bin, p_b is approximated as the area of a rectangle where the height is the total posteriori probabilities which fall into that region. Then, the entropy is approximated as

$$\mathbb{H} \approx -\sum_{b=1}^{B} p_b \log \frac{p_b}{w_b}.$$

In Fig. 7, the histograms computed at each base-layer decision space and the fusion space are displayed for test dataset. It is observed that the uncertainties of the distributions are decreased in the fusion space.

TABLE XIV: Entropy values computed in feature spaces for test set.

Feature Spaces	Class 1	Class 2	Class 3	Class 4
Homogeneous Texture	0.3751	0.3840	0.3702	**0.0679**
Color Layout	0.1905	**0.2644**	0.3255	0.0861
MFCC	0.1920	0.3824	**0.0879**	0.3347
Chromagram	0.3442	0.3621	0.2011	0.2834
Fluctuation	**0.0389**	0.3013	0.3115	0.4276

Entropy values given in Table XIV provide information about the data uncertainty in the feature spaces. It is expected that a classifier employed on F_j with relatively lower entropy for a particular class ω_c classifies the samples belonging to ω_c with a better performance than the samples belonging to other classes.

For instance, distributions of Fluctuation, MFCC and Homogeneous Texture features have

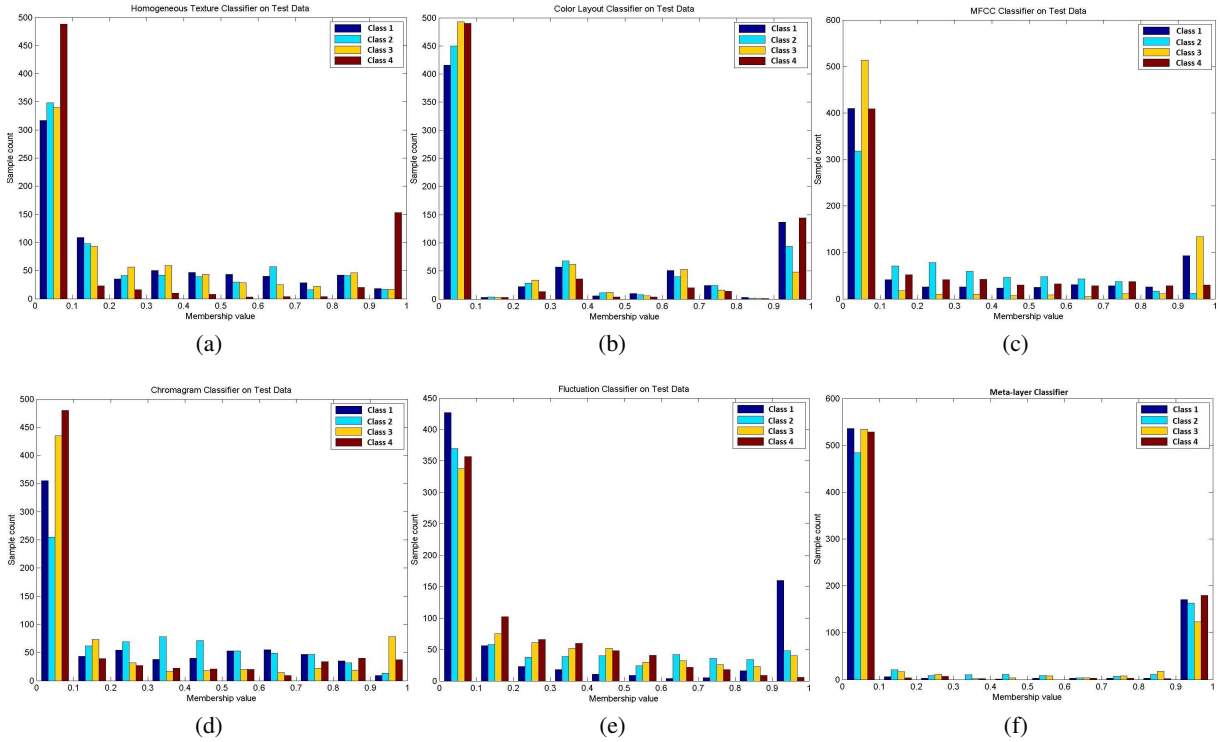


Fig. 7: Histograms which represent distributions computed in the individual decision spaces of base-layer classifiers employed using (a) Histogram Texture, (b) Color Layout, (c) MFCC, (d) Chromagram, (e) Fluctuation features, and (f) in the fusion space of the meta-classifier in FSG. Notice that the lowest entropy is observed in the fusion space.

TABLE XV: Entropy values computed in decision and fusion spaces for test set.

Decision and Fusion Spaces	Class 1	Class 2	Class 3	Class 4
Homogeneous Texture	0.2160	0.2360	0.2550	**0.0457**
Color Layout	0.1057	0.3052	0.2383	0.4584
MFCC	0.1539	0.2161	**0.1322**	0.1936
Chromagram	0.1165	**0.1092**	0.1582	0.1760
Fluctuation	**0.0344**	0.2286	0.2890	0.3228
Fusion Space	*0.0228*	*0.0529*	*0.0873*	*0.0156*

the lowest entropy values for the first, third and fourth classes, respectively (see Table XIV). The base-layer classifiers which use these features provide the highest classification performances as shown in Table XI.

Although the distribution of Color Layout features provides the lowest entropy for the second class, the base-layer classifier employed on Color Layout features performs worse than the other classifiers. However, the features of the samples belonging to the fourth class

have the lowest entropy in Color Layout space (see; Table XIV). As a result, the classifier employed on Color Layout space gives the highest classification performance for the fourth class as given in Table XI.

Entropy values computed in decision and fusion spaces are given in Table XV for test dataset. Entropy values computed in decision spaces represent the decision uncertainty of base-layer classifiers for each class. Note that the classifiers employed on the feature spaces with minimum decision uncertainties for particular classes provide the highest classification performances for these classes (see Table XI). Entropy values of the membership vectors $\bar{\mu}(\bar{x}_i)$ that reside in the fusion space represent the joint entropy of $\{\bar{\mu}(\bar{x}_{i,j})\}_{j=1}^{J}$, since $\bar{\mu}(\bar{x}_i) = [\bar{\mu}(\bar{x}_{i,1}) \dots \bar{\mu}(\bar{x}_{i,j}) \dots \bar{\mu}(\bar{x}_{i,J})]$, $\forall i = 1, 2, \dots, N$. If the classifier decisions are independent, then the entropy value Ent_{fusion} of $\bar{\mu}(\bar{x}_i)$ is equal to the sum of the entropy values Ent_j of $\bar{\mu}(\bar{x}_{i,j})$, $\forall i = 1, 2, \dots, N$, such that

$$Ent_{fusion} = \sum_{j=1}^{J} Ent_j.$$

However, we observe that $Ent_{fusion} \leq \sum_{j=1}^{J} Ent_j$ in Table XV, which implies that the decisions are dependent. This dependency is attributed to the shareability of the samples among the classifiers in the FSG as shown in Table XIII. Thereby, lower entropy values are obtained in the fusion space.

VIII. Summary and Conclusion

In this study, the distance learning problem of a single classifier is extended to formalize a decision fusion problem of an ensemble of classifiers. This task is achieved by minimizing the difference between the N-sample and large-sample classification error of the nearest neighbor classifier.

The classification error is minimized by a distance learning algorithm of a decision fusion method, called Fuzzy Stacked Generalization (FSG). For this purpose, the distance learning problem is reformulated as a feature space, decision space and fusion space design problem of the FSG. The base-layer classifiers of the FSG are used for two purposes; i) mapping the feature vectors to decision vectors and ii) estimating posterior probabilities of base-layer

classifiers, which are the variables of the distance function. Decision vectors, which represent the posterior probabilities in the decision spaces, are then concatenated to construct the feature vectors in the fusion space of a meta-layer classifier. Finally, the vectors residing in the fusion space are used to minimize the distance between the N-sample and large-sample errors by a meta-layer fuzzy k-NN classifier.

The rationale behind using the fuzzy k-NN method in the base-layer classifiers of FSG is many folded. First of all, fuzzy k-NN is a *powerful* nonparametric density estimation method used for the estimation of posterior probabilities, which are crucial in designing distance functions. Second, the error of k-NN is upper and lower bounded by the Bayes Error which is the minimum achievable classification error by any classification algorithm. Therefore, one of the major contributions of the suggested decision fusion method is to minimize the difference between N-sample and large-sample classification error of k-NN to bridge the gap between the N-sample classification error of k-NN and Bayes error. This task is achieved by using the distance learning approach of Short and Fukunaga [23].

The proposed FSG algorithm is tested on artificial and benchmark datasets and the results are compared to the state of the art algorithms, such as Adaboost, Rotation Forest and Random Subspace.

In the experiments on artificial datasets, it is observed that if the dataset is *shareable* by the base-layer classifiers, then the classification performance of FSG gets significantly higher than that of the individual base-layer classifiers. The experiments show that the performance of FSG depends on the degree of collaboration among the classifiers to correctly recognize the features of the samples, rather than the performance of each individual classifier.

In the experiments on benchmark datasets, the proposed FSG algorithm outperforms the state of the art algorithms, basically because of two reasons: First, the proposed FSG algorithm bounds the dimension of the feature vectors in the fusion space to CJ (number of classes $*$ number of feature extractors) no matter how high is the dimension of the individual feature vectors of the base-layer classifiers. This property of the FSG avoids the curse of dimensionality problem. Second, employing distinct feature extractors for each base-layer classifier enables us to split various attributes of the feature spaces. Therefore, each base-layer classifier gains an expertise to learn a specific property of a sample, and correctly

classifies a group of samples belonging to a certain class in the training data. This approach assures the diversity of the classifiers as suggested by Kuncheva [48], [20] and enables the classifiers to collaborate for learning the classes or groups of samples. It also allows us to optimize the parameters of each individual base-layer classifier independent of the other.

In the experiments on the multi-modal dataset, even if the performances of the individual base-layer classifiers are low for some classes, the performance of the meta-layer classifier of the FSG is boosted significantly. Moreover, it is observed that the entropies of distributions of features are decreased through the feature space transformations from the base-layer to the meta-layer of the architecture. Therefore, the FSG architecture transforms the high dimensional and linearly non-separable feature spaces of the base-layer classifiers into a relatively more separable fusion space with fixed dimension.

ACKNOWLEDGEMENT

M. Ozay was supported by the European commission project PaCMan EU FP7-ICT, 600918.

REFERENCES

[1] D. H. Wolpert, "Original contribution: Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, Feb 1992.

[2] N. Ueda, "Optimal linear combination of neural networks for improving classification performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 2, pp. 207–215, Feb 2000.

[3] M. U. Sen and H. Erdogan, "Linear classifier combination and selection using group sparse regularization and hinge loss," *Pattern Recogn. Lett.*, vol. 34, no. 3, pp. 265–274, 2013.

[4] N. Rooney, D. Patterson, and C. Nugent, "Non-strict heterogeneous stacking," *Pattern Recogn. Lett.*, vol. 28, no. 9, pp. 1050–1061, 2007.

[5] B. Zenko, L. Todorovski, and S. Dzeroski, "A comparison of stacking with meta decision trees to bagging, boosting, and stacking with other methods," in *Proceedings of the 2001 IEEE International Conference on Data Mining*, ser. ICDM '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 669–670.

[6] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Recognizing partially occluded, expression variant faces from single training image per person with som and soft k-nn ensemble," *IEEE Trans. Neural Netw.*, vol. 16, no. 4, pp. 875–886, Jul 2005.

[7] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *J. Artif. Int. Res.*, vol. 10, no. 1, pp. 271–289, May 1999.

[8] R. E. Schapire, "A brief introduction to boosting," in *Proceedings of the 16th international joint conference on Artificial intelligence - Volume 2*, ser. IJCAI'99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 1401–1406.

[9] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug 1998.

[10] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, Oct 2006.

[11] A. Ghorbani and K. Owrangh, "Stacked generalization in neural networks: generalization on statistically neutral problems," in *IEEE International Joint Conference on Neural Networks*, vol. 3, 2001, pp. 1715–1720.

[12] G. Zhao, Z. Shen, C. Miao, and R. Gay, "Enhanced extreme learning machine with stacked generalization," in *IEEE International Joint Conference on Neural Networks*, 2008, pp. 1191–1198.

[13] M. Ozay and F. T. Vural, "On the performance of stacked generalization classifiers," in *Proceedings of the 5th international conference on Image Analysis and Recognition*, ser. ICIAR '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 445–454.

[14] E. Akbas and F. T. Yarman Vural, "Automatic image annotation by ensemble of visual descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[15] G. Sigletos, G. Paliouras, C. D. Spyropoulos, and M. Hatzopoulos, "Combining information extraction systems using voting and stacked generalization," *J. Mach. Learn. Res.*, vol. 6, pp. 1751–1782, Dec 2005.

[16] M. Ozay and F. T. Yarman Vural, "A new decision fusion technique for image classification," in *Proceedings of the 16th IEEE the International Conference on Image Processing, (ICIP 2009)*, Cairo, Egypt, Nov 2009, pp. 2189–2192.

[17] S. Džeroski and B. Ženko, "Is combining classifiers with stacking better than selecting the best one?" *Mach. Learn.*, vol. 54, no. 3, pp. 255–273, Mar 2004.

[18] S.-B. Cho and J. H. Kim, "Multiple network fusion using fuzzy logic," *IEEE Trans. Neural Netw.*, vol. 6, no. 2, pp. 497–501, Mar 1995.

[19] L. I. Kuncheva, ""fuzzy" versus "nonfuzzy" in combining classifiers designed by boosting," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 729–741, Dec 2003.

[20] ——, *Fuzzy Classifier Design*, ser. Studies in Fuzziness and Soft Computing. Springer, 2000, vol. 49.

[21] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 683–697, Sep 1992.

[22] K. E. Graves and R. Nagarajah, "Uncertainty estimation using fuzzy measures for multiclass classification," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 128–140, Jan 2007.

[23] R. D. S. II and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 622–626, 1981.

[24] E. Marchiori, "Class conditional nearest neighbor for large margin instance selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 364–370, Feb 2010.

[25] ——, "Hit miss networks with applications to instance selection," *J. Mach. Learn. Res.*, vol. 9, pp. 997–1017, Jun 2008.

[26] Y. Li and L. Maguire, "Selecting critical patterns based on local geometrical and statistical information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 6, pp. 1189–1201, Jun 2011.

[27] S. Garcia, J. Derrac, J. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 417–435, Mar 2012.

[28] F. Fernandez and P. Isasi, "Local feature weighting in nearest prototype classification," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 40–53, 2008.

[29] J. Derrac, I. Triguero, S. Garcia, and F. Herrera, "Integrating instance selection, instance weighting, and feature weighting for nearest neighbor classifiers by coevolutionary algorithms," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 5, pp. 1383–1397, 2012.

[30] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 513–520.

[31] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Jun 2009.

[32] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng, "Online and batch learning of pseudo-metrics," in *Proceedings of the Twenty First International Conference on Machine learning*, ser. ICML '04. New York, NY, USA: ACM, 2004, pp. 94–101.

[33] R. Paredes and E. Vidal, "Learning weighted metrics to minimize nearest-neighbor classification error," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1100–1110, 2006.

[34] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York, NY, USA: Wiley, 2001.

[35] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan 1967.

[36] E. Fix and J. L. Hodges, "Discriminatory analysis — nonparametric descrimination: consistency properties," USAF School of Aviation Medicine, Randolph Field, Texas, Report 4, 1951, project No. 21-29-004.

[37] J. Keller, M. Gray, and J. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Trans. Syst., Man, Cybern., Syst*, vol. SMC-15, no. 4, pp. 580 –585, 1985.

[38] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working set selection using second order information for training support vector machines," *J. Mach. Learn. Res.*, vol. 6, pp. 1889–1918, Dec 2005.

[39] C. B. D. Newman and C. Merz, "UCI repository of machine learning databases," 1998. [Online]. Available: http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html

[40] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 221–228.

[41] V. Garcia, E. Debreuve, F. Nielsen, and M. Barlaud, "k-nearest neighbor search: fast GPU-based implementations and application to high-dimensional feature matching," in *IEEE International Conference on Image Processing (ICIP)*, Hong Kong, China, September 2010.

[42] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[43] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 607–616, Jun 1996.

[44] H. Eidenberger, "Statistical analysis of content-based mpeg-7 descriptors for image retrieval," *Multimedia Syst.*, vol. 10, no. 2, pp. 84–97, 2004.

[45] P. Salembier and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*, B. Manjunath, Ed. New York, NY, USA: John Wiley & Sons, Inc., 2002.

[46] O. Lartillot and P. Toiviainen, “A matlab toolbox for musical feature extraction from audio,” in *Proceedings of the 10th International Conference on Digital Audio Effects*, Bordeaux, France, Sep 2007, pp. 237–244.

[47] K. F. Wallis, “A note on the calculation of entropy from histograms,” Department of Economics, University of Warwick, UK, Tech. Rep., 2006.

[48] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, 1st ed. Haboken, NJ, USA: Wiley-Interscience, 2004.