Exploiting Object Similarity for Robotic Visual Recognition

Hong Cai and Yasamin Mostofi

***Abstract*—In this paper, we are interested in robotic visual object classification using a Deep Convolutional Neural Network (DCNN) classifier. We show that the correlation coefficient of the automatically-learned DCNN features of two object images carries robust information on their similarity, and can be utilized to significantly improve the robot's classification accuracy, without additional training. More specifically, we first probabilistically analyze how the feature correlation carries vital similarity information and build a Correlation-based Markov Random Field (CoMRF) for joint object labeling. Given query and motion budgets, we then propose an optimization framework to plan the robot's query and path based on our CoMRF. This gives the robot a new way to optimally decide which object sites to move close to for better sensing and for which objects to ask a remote human for help with classification, which considerably improves the overall classification. We extensively evaluate our proposed approach on 2 large datasets (e.g., drone imagery, indoor scenes) and several real-world robotic experiments. The results show that our proposed approach significantly outperforms the benchmarks.**

***Index Terms*—Object Detection, Segmentation, and Categorization; Deep Learning in Robotics and Automation; AI-Based Methods; Co-Optimization of Robotic Path Planning, Querying, and Visual Recognition**

I. Introduction

Consider a robotic operation where a robot is tasked with visual sensing and classification in an area, an example of which is shown in Fig. 1. The robot takes several object images with its camera and uses a trained Deep Convolutional Neural Network (DCNN) classifier to label them to a given set of classes. If its classification confidence is low for some of the images, it then has to plan how to use its limited resources (e.g., motion energy budget, queries) to move closer to some of the object locations for better sensing, and/or to ask a remote human operator to help with the classification. Such visual task scenarios are commonly seen in real-world robotic applications, such as scene understanding, search and rescue, and surveillance.

In this paper, we are interested in enhancing the performance of robotic visual classification by exploiting the similarity structure among the robot's visual inputs. We show that such similarity structure can be freely and robustly inferred from the output of a trained DCNN classifier. More specifically, we show that the Pearson correlation coefficient of the feature vectors of two object images from the output of a trained DCNN classifier carries reliable information on the similarity of the two corresponding objects (i.e., do they belong to the same class?), even if the individual classification accuracy of each object was not very high. This then allows us to design a Markov Random Field (MRF)-based joint labeling framework, where the similarity information is utilized to reduce the classification uncertainty. For instance, in Fig. 1, there are two images (2 and 3) of the same class (person) but in different poses. The robot's vision initially misclassifies them to a truck and a car. However, if the robot is aware that they belong to the same class (while it cannot properly classify them), then it can correctly classify both to persons using our MRF joint labeling framework. We then show the implication of this correlation-based joint labeling for the robot's field decision-making by co-optimizing its query, path, and visual labeling. We next discuss the state of the art in vision, robotics, and machine learning, as related to this paper.

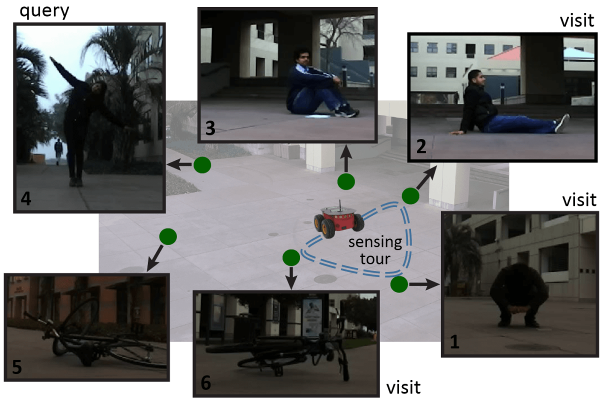


Fig. 1: A robot is tasked with classifying objects on our campus. It may have low confidence in some of its initial classifications. Given a limited human query budget and a motion budget, the robot then needs to decide which object locations it should visit to sense better, and which object images it should ask a remote human operator to classify, in order to improve its overall classification performance. By using the correlation coefficient of the feature vectors from a trained DCNN classifier, the robot can robustly capture image similarities (e.g., objects 2 and 3 belong to the same class), which has a significant implication for its field decision-making and joint labeling, as we show in this paper. Readers are referred to the color pdf to better view the images in this paper.

State of the Art: In computer vision, machine learning and deep neural networks have significantly advanced the state of the art, in areas such as detection [1], segmentation [2], and DCNN architectural design [3]–[6]. While most research efforts have focused on making the machine better at processing individual visual inputs, relationship among a number of visual inputs can also be exploited to design better vision systems, as some recent papers show. For instance, Galleguillos et al. [7] utilize object co-occurrence and spatial context to design categorization algorithms. Torralba et al. [8] study place-object co-recognition, where the semantic consistency between an object and the current place is taken into account. In image segmentation, pixel-level relationship is utilized in the classification of each pixel [2]. Spatial-temporal

Hong Cai and Yasamin Mostofi are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, USA (email: {hcai, ymostofi}@ece.ucsb.edu). This work was supported in part by NSF RI award #1619376 and NSF CCSS award #1611254.

correlation has also been considered in video applications [9], [10]. More related to our work are those that consider object/image similarity. Several papers have proposed DCNN-based similarity metrics, which require dedicated training to learn a similarity measure for a specific application (e.g., patch matching, image retrieval, image error assessment) [11]–[15]. A few papers have explored off-the-shelf DCNN features for applications such as clustering [16], [17]. As for visual detection and classification, a few papers have utilized pairwise object/image similarity to improve the recognition accuracy [18]–[20]. However, these methods use simple hand-crafted features (e.g., color histogram), which will not work well in complex visual tasks that can involve many object classes, different poses/views of the same object class, and/or images degradations.[1]

The robotics community has also started to exploit the relationship among visual inputs for robotic vision applications. For instance, for object labeling, Koppula et al. [21] incorporate geometric context among objects, and Ali et al. [22] utilize the co-occurrence relationship. Ruiz et al. [23] utilize semantic knowledge to derive compatibilities among objects and rooms for joint recognition. However, these methods require additional training in order to use these contextual relations. For instance, extensive training is needed to utilize geometric relations for joint labeling in [21].

In some cases, the machine can obtain a few ground-truth labels via querying to improve its performance. Most related to this work are those that optimize the query selection based on a correlation model (e.g., an MRF). In [24], Krause et al. utilize information theoretic metrics (e.g., mutual information) to select queries most beneficial to labeling the remaining unqueried instances. Recently, Wang et al. [25] use Bayesian lower bounds to optimize the selection, which outperforms the earlier information theoretic methods. Another related subject is active learning, which studies how to select labeled samples to better train a learning algorithm (e.g., [9], [26], [27]). The formulation of active learning, however, is different from our problem, as we do not consider retraining the vision algorithm during deployment in this work. In the context of robotics, several papers have studied how to optimize the robot's motion to acquire more information and improve its visual sensing [28]–[30]. These existing papers, however, do not take into account the correlation among the visual targets.

As discussed above, while several types of relationship between visual inputs have been exploited in robotic vision, object similarity has not been exploited in robotics. **It is our hypothesis that the similarity between two visual inputs can be inferred robustly from the output of a trained DCNN classifier, without any additional training.** This would then have a significant implication for the robot's visual classification, and its field decision-making in terms of visual sensing, path planning, and querying, as we shall show in this paper. We next explicitly discuss the contributions of the paper.

Statement of Contributions:

1. We probabilistically analyze the correlation coefficient between the features of two images from a trained DCNN classifier in an extensive study based on $180,000$ image pairs from 39 classes, for 3 commonly-used state-of-the-art DCNN architectures. We show that the correlation coefficient can capture pairwise image similarity robustly, even when the images are subject to low illumination and low resolution, or are misclassified. This similarity measure comes for free from the DCNN classifier, requiring no additional training.

2. Based on the probabilistic characterization of this pairwise image similarity metric, we build a correlation-based MRF (CoMRF) for joint labeling, which allows the robot to better label the objects based on the correlation structure. Given query and motion budgets, we propose a CoMRF-based query-motion co-optimization approach to jointly plan the robot's query and path. This allows the robot to optimally decide which objects it should visit for better sensing, and for which visual inputs it should ask for human help. As we shall see, by utilizing the proposed framework, the robot can improve its visual performance significantly, under the motion and query budgets. In other words, our proposed method reduces the robot's motion and query/communication burdens for labeling, while achieving the same task quality.

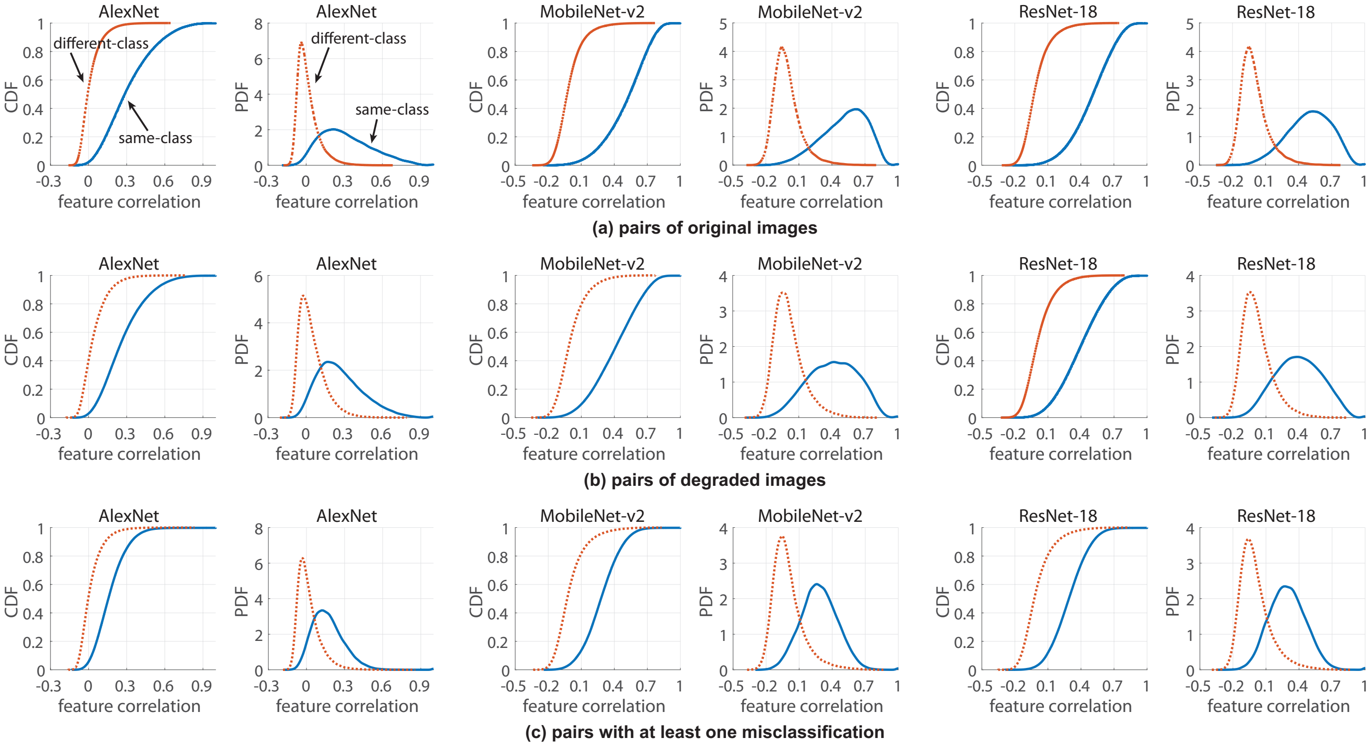
3. By using 1) a large COCO-based test set, 2) the challenging large-scale drone imagery dataset of VisDrone, and 3) the large indoor scene dataset of NYU-v2, we extensively evaluate our proposed approach on joint labeling, query selection, and path planning, across a large variety of realistic scenarios. The results show that our proposed approach significantly outperforms the state of the art (e.g., outperforms [20] by 0.240 in terms of classification accuracy). We then run several real-world robotic experiments to further demonstrate the efficacy of our proposed CoMRF-based query-motion co-optimization algorithm. The results verify that our approach considerably outperforms the benchmark.

The rest of the paper is organized as follows. In Sec. II, we introduce our object similarity metric and confirm its reliability based on extensive studies. We further propose CoMRF for joint labeling. In Sec. III, we develop our query-motion co-optimization algorithm based on our CoMRF. In Sec. IV and V, we evaluate our proposed approach on joint labeling, query selection, and path planning, on a large COCO-based test set and a large-scale drone imagery dataset, respectively. In Sec. VII, we further demonstrate the efficacy of our proposed algorithm with several robotic experiments. We discuss a few more aspects of our methodology in Sec. VIII and finally conclude in Sec. IX.

II. Correlation-based Markov Random Field

In this section, we first establish that the correlation coefficient between two feature vectors, from a DCNN-based classifier, provides a reliable metric for characterizing the probability that the two images belong to the same class. We then show how to build a correlation-based Markov Random Field (CoMRF) for joint object labeling, which captures and utilizes our pairwise probabilistic object similarity metric.

[1] Note that in order to classify the objects in a scene, a robot would first need to extract the relevant object regions from the acquired image. An end-to-end detection network is typically utilized in such cases, which performs both localization and classification simultaneously. However, the state-of-the-art detection networks can miss objects when they are visually challenging. As such, we utilize a saliency and depth-based localization algorithm in our experiments, as we shall see in Sec. VII.



(a) pairs of original images

(b) pairs of degraded images

(c) pairs with at least one misclassification

Fig. 2: (a) In each CDF-PDF figure pair of a DCNN classifier (e.g., the two leftmost figures of AlexNet), the left figure shows the CDFs of the feature correlation of same-class (blue solid) and different-class (red dashed) pairs, and the right figure shows the PDFs of the feature correlation of same-class and different-class pairs. (b) CDFs and PDFs of the feature correlation after degrading the images by low resolution and low illumination. (c) CDFs and PDFs of the feature correlation with at least one misclassification in each image pair.

A. *Image Similarity and Feature Correlation*

Consider a pair of images, each containing an object of interest. The images have passed through a trained DCNN classifier, which automatically provides a feature vector for each image. In contrast to hand-crafted features, this automated feature vector contains robust information on the essence of an object, which allows for capturing image similarity reliably, without additional training, as we show in this section.

To test our hypothesis about this image similarity metric, we construct a large image classification dataset, which contains 39 object classes, including a variety of daily objects (e.g., person, car). There are $76,505$ images in total, collected from the COCO dataset [31] and ImageNet [32]. Most of the images are obtained from the COCO detection dataset by extracting object image patches based on the provided bounding box annotations, in order to better represent what the robot would see in real-world classification tasks (e.g., images that can be small, have occlusion/clutter, and have non-ideal lighting/contrast). We divide this dataset into $38,555$ training images, $19,350$ validation images, and $18,600$ test images.[2] Utilizing this dataset, we have trained DCNN classifiers using the following three commonly-used state-of-the-art architectures: AlexNet [3], MobileNet-v2 [5], and ResNet-18 [6], with their respective accuracies over the validation set as follows: 0.800, 0.873, and 0.873. We refer to these as the base classifiers in the paper. By utilizing these three different network architectures, we will establish the generalizability of our metric to different networks. Each trained DCNN classifier then automatically provides a feature vector for an input image from the layer prior to the final output layer. For instance, in the case of AlexNet, this feature vector is provided by the activations on the 7^{th} layer, which is a fully-connected layer prior to the final output layer. The test set is reserved for evaluating our proposed methodology in Sec. IV.

***Remark** 1:* We note that we could have used an existing trained classifier to test our image similarity metric. However, we chose to train a classifier in order to train on classes more relevant to the types of objects that an unmanned vehicle may see (e.g., people, bikes, cars). The unmanned vehicle will then use this classifier for our real-world experimental tests in Sec. VII. We thus emphasize that the aforementioned training is simply to train a classifier and our proposed image similarity metric is freely available from any DCNN classifier.

During the training of a DCNN classifier, the network learns relevant features to feed to the output layer, which is a linear operator. As such, the DCNN is trained to derive features to linearly separate images of different classes. Motivated by this generic design of DCNN classifiers, we utilize the Pearson linear correlation coefficient to measure the similarity between a pair of images. More specifically, given a feature vector for each image in a pair of images, we compute the Pearson linear correlation coefficient of the feature vectors of the two images. We refer to this metric as the **feature correlation**. We next empirically analyze its distribution. Fig. 2 (a) shows the Cumulative Distribution Functions (CDFs) and Probability Density Functions (PDFs) of the feature correlation of $180,000$ random image pairs from the validation set, based on the

[2] Detailed descriptions of this dataset can be found in Appendix A.

Base classifier	Correlation threshold	Prob. above threshold	
		different-class	*same-class*
AlexNet	0.3	0.009	0.473
MobileNet-v2	0.4	0.009	0.720
ResNet-18	0.4	0.009	0.693

TABLE I: Correlation thresholds for the three DCNN classifiers. The third and fourth columns show the probability of false correlation (different-class declared as same-class) and the probability of a same-class pair having a correlation above the threshold, respectively.

features provided by the trained AlexNet, MobileNet-v2, and ResNet-18, respectively. More specifically, in each CDF-PDF pair (e.g., the two leftmost figures of AlexNet in Fig. 2 (a)), the blue solid curve of the left figure shows the CDF of the feature correlation of a pair of object images, given these two objects belong to the same class, which we denote as a **same-class pair**. The red dashed curve then shows the CDF of the feature correlation of a pair of objects belonging to different classes, which we denote as a **different-class pair**. It can be seen that, for each DCNN classifier, there is a considerable difference between these two distributions. For a different-class pair, the feature correlation is more likely to be small, while for a same-class pair, there is a higher chance of a high correlation. Similarly, the PDF curves show that the distributions of the correlation coefficient are well separated for same-class and different-class pairs. This further motivates utilizing feature correlation to deduce whether two objects belong to the same class. To do so, we need a threshold, above which to declare two objects in the same class and below which to declare otherwise. We choose a threshold such that the probability of false correlation (different-class declared as same-class) is very small. For instance, for AlexNet, we can see that the probability of a different-class pair having a correlation above 0.3 is less than 0.010, while 47.3% of the same-class pairs have a correlation above 0.3. We then use 0.3 as our threshold for AlexNet in our experiments. Similarly, we select thresholds for the MobileNet-v2 and ResNet-18 classifiers such that the probability of false correlation is very small, while a large percentage of same-class pairs still have a correlation above the threshold. This allows us to capture many of the same-class objects, while ensuring a very small probability of mistaking a different-class pair for a same-class pair. The respective thresholds for the three DCNN classifiers are summarized in Table I, along with the probability of false correlation and the probability of same-class pairs having a correlation above the threshold. When two images have a high feature correlation, there is a high probability that they belong to the same class. Therefore, when designing our CoMRF in Sec. II-B, every two images with a feature correlation above the threshold are connected by an edge and are more likely to have the same label in the joint labeling process.

Robustness to Visual Differences: As two instances of the same object class can be very different visually, it is important that our image similarity metric declares many of such instances to be in the same class. We extensively study our dataset from this angle, in order to ensure that it contains a variety of poses/views for each object class. The PDF/CDF plots of Fig. 2 (a), for instance, are obtained from a general dataset (the validation part of our constructed dataset) with visually-diverse instances of each object class. Fig. 3 shows 10 random samples of bicycle for instance. Using the AlexNet classifier and based on the threshold of 0.3, a link is drawn between two objects if their feature correlation is above 0.3. As can be seen, despite the drastic visual differences, the resulting graph is dense (55.56% of all possible pairwise links are captured), indicating that the feature correlation robustly captures same-class objects.

Robustness to Image Degradation: In practice, robot sensing may suffer from various degradations, such as low resolution and low illumination. As our approach relies on the feature correlation of image pairs, it is important to understand its robustness to image degradation. In other words, the difference between the feature correlation distributions of same-class and different-class pairs should still be large enough. To evaluate this, we have corrupted the validation set images by randomly reducing resolution and illumination. Fig. 2 (b) shows the CDF/PDF of the feature correlation for the corrupted image dataset and the three network architectures. As can be seen, although the difference between the two distributions has become smaller, they are still robustly different.

Robustness to Misclassification: **The feature correlation can identify same-class objects, even when the classifier misclassifies them.** This is important as when there are misclassified images, the feature correlation should still robustly capture the similarity, which can then be utilized to correct the misclassifications. The initial image pool used to plot Fig. 2 (a) includes several image pairs with at least one misclassified image. In order to more explicitly show the robustness of the feature correlation to misclassification, Fig. 2 (c) shows the feature correlation distributions of same-class and different-class pairs when at least one of the images in a pair is misclassified. It can be seen that, for each DCNN classifier, using the corresponding threshold, the feature correlation still captures many same-class pairs with a small false correlation probability (less than 0.018 for all three DCNN classifiers) in this challenging setting. Fig. 1 shows an example of this where, using the AlexNet classifier, the two human images (2 and 3) are initially misclassified as a truck and a car. Utilizing the feature correlation, the robot can infer that they are highly likely to be in the same class, which allows it to jointly label them correctly, using the method we propose in Sec. II-B.[3]

B. Correlation-based Markov Random Field (CoMRF)

So far, we have established that feature correlation provides a reliable metric for image similarity. We next show how it can be used for joint labeling. Suppose that we have N images, each containing an object-of-interest (defined as an object belonging to the set of classes with which the classifier was trained). We next construct our CoMRF based on the pairwise feature correlations.[4]

[3] Even when the robot's initial classifications are correct, it may have low confidence on some of the images. In such cases, the robot can still benefit tremendously from our similarity metric, which can be utilized to reduce the uncertainty in the robot's classification decisions using our CoMRF method of Sec. II-B.

[4] Readers are referred to the MRF literature for more details on the terminology (e.g., [33], [34]).

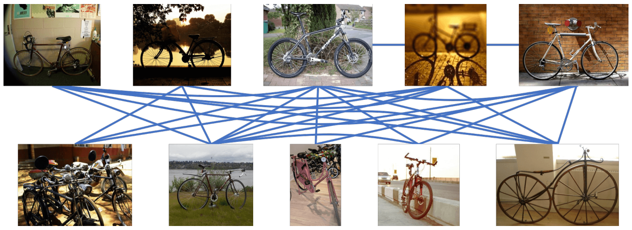


Fig. 3: Robustness of our metric to visual variations – visualization of the feature correlation among 10 random bicycle images. Two images are connected if their AlexNet-based feature correlation is above the threshold. The number of edges in this graph is 55.56% of that of a 10-node complete graph, indicating that many of the pairs are correctly declared as same-class despite drastic visual differences.

In our MRF, each object image is represented by a node and two nodes are connected by an edge if their pairwise feature correlation is above a certain threshold s_{T} (e.g., the threshold for the AlexNet classifier is 0.3, as discussed in Sec. II-A). The overall potential function of the CoMRF is then given as follows:

$$P(x_1,...,x_N)=\prod_{i=1}^{N}\phi_i(x_i)\prod_{(i,j)\in L}\psi(x_i,x_j), \tag{1}$$

where $x_i\in\{1,...,N_c\}$ is the label variable of the i^{th} image with N_c denoting the total number of object classes, $\phi_i(x_i)$ is the node potential function, $L=\{(i,j)|s_{i,j}\geq s_T\}$ is the set of pairs with feature correlation $s_{i,j}$ above s_{T}, and $\psi(x_i,x_j)$ is the edge potential function.

Node Potential: For each node i, $\phi_i(x_i)=p_{X_i}(x_i)$, where $p_{X_i}(x_i)$ is the probability distribution over the classes from the classifier's output and $x_i\in\{1,...,N_c\}$ is the label variable.

Edge Potential: Given that there is an edge between nodes i and j, we denote the probability that these two nodes have the same label as $p_{\mathrm{same}}=p(x_i=x_j|s_{i,j}\geq s_{\mathrm{T}})$, where $s_{i,j}$ is the feature correlation between nodes i and j. If $x_i=x_j$, we assume that it is equally probable for nodes i and j to belong to any one of the N_c classes. Similarly, if $x_i\neq x_j$, then we assume that it is equally probable for nodes i and j to take any pairwise combination of the N_c classes. The edge potential function is then constructed as follows: $\psi(x_i,x_j)=p_{\mathrm{same}}/N_c$, if $x_i=x_j=k, \forall k\in\{1,...,N_c\}$, and $\psi(x_i,x_j)=(1-p_{\mathrm{same}})/(N_c^2-N_c)$, if $x_i\neq x_j, \forall x_i,x_j\in\{1,...,N_c\}$.

In a specific joint labeling task instance, given that the objects present belong to $N_p\leq N_c$ classes (subset out of the total N_c classes seen in training) and the threshold is s_{T}, p_{same} can be written as follows using the Bayes rule:

$$\begin{aligned}&p_{\mathrm{same}}=p(x_i=x_j|s_{i,j}\geq s_{\mathrm{T}})\\&=\frac{p(s_{i,j}\geq s_{\mathrm{T}}|x_i=x_j)p(x_i=x_j)}{p(s_{i,j}\geq s_{\mathrm{T}}|x_i=x_j)p(x_i=x_j)+p(s_{i,j}\geq s_{\mathrm{T}}|x_i\neq x_j)p(x_i\neq x_j)},\end{aligned}$$

where $p(x_i=x_j)=N_p/N_p^2$ and $p(x_i\neq x_j)=N_p(N_p-1)/N_p^2$ are the prior probabilities that two nodes belong to the same class and different classes, respectively.

Since this is dependent on N_p, which the robot does not know during the operation, we average p_{same} over the distribution of N_p. For instance, in practice, we do not expect N_p to be larger than 10 in a task instance. Thus, we assume that N_p is uniform over $\{1,...,10\}$, and numerically evaluate $E[p_{\mathrm{same}}]$ to be 0.928, 0.904, and 0.878 for AlexNet, MobileNet-v2, and ResNet-18, respectively. For the rest of the paper, for each DCNN classifier, we then set $p_{\mathrm{same}}=E[p_{\mathrm{same}}]$ in our CoMRF implementation. In order to compute the posterior distribution of the nodes, we use Loopy Belief Propagation (LBP), which is an approximate inference algorithm [33]. The final estimated label for a node is then given by $\hat{x}_i=\operatorname{argmax}\tilde{p}_{X_i}(x_i)$, where $\tilde{p}_{X_i}$ is the posterior marginal distribution of node i over the N_c classes, after running LBP on CoMRF.[5]

III. Optimization of Querying and Path Planning

Consider the case where the robot is tasked with object classification in an area. The robot does an initial classification based on visual sensing and the state-of-the-art DCNN classifier. However, its classification confidence may not be high for several objects. The robot is given a query budget to ask for human help and/or a motion budget to move to some of the object locations to sense better. In this section, we propose our methodology for co-optimizing query selection and path planning, based on the CoMRF. More specifically, when the robot visits a site to better sense an object, or queries the human, it can obtain the correct label of the corresponding object with a high probability. Given these new labels, the robot can perform conditional inference over CoMRF and update all the remaining labels. Thus, the robot's sensing and query directly affect its joint labeling.

Ideally, the robot should select a subset of nodes to query and/or visit such that the posterior joint uncertainty of all the nodes on the MRF is minimized. However, the computational complexity of finding the optimum to this problem is very high, and existing methods either resort to greedy schemes or are limited to chain-structure graphs [24], [36]. Therefore, we instead consider a **neighborhood uncertainty measure** that can still capture the feature correlation and object similarity. More specifically, for each node, we propose an uncertainty measure that takes into account both its individual uncertainty and the uncertainty of its neighboring nodes in CoMRF. This is because as the correct label is applied to a node, not only is its own uncertainty eliminated, but the uncertainty of the neighbors will also be reduced. This measure then provides a way to quantify the amount of (neighborhood) uncertainty reduction, should a node be given the correct label. Our uncertainty measure is then as follows, for each node i,

$$r_i=(1-c_i)+w\sum_{j\in A(i)}(1-c_j),\quad \forall i\in\{1,...,N\}, \tag{2}$$

where $c_i=\max\tilde{p}_{X_i}(x_i)$, with $\tilde{p}_{X_i}$ being the posterior marginal distribution of node i (after LBP), $x_i\in\{1,...,N_c\}$ is the label variable. $1-c_i$ is then a measure of the individual uncertainty of node i. The second term measures the uncertainty of the neighboring nodes of node i, where $A(i)$ denotes the neighbor set of node i. $w\geq 0$ weighs the respective importance of the individual and neighborhood uncertainties.[6]

Given this uncertainty measure, we next formulate our query selection and path planning co-optimization, which selects

[5] We use the publicly available MRF library [35] for our implementation.

[6] We set $w=5$ in our implementation, based on running AlexNet on the validation set. The weight $w=5$ is then used for all three DCNN classifiers for all the experiments in Sec. IV, V, and VII

Algorithm 1: Proposed query selection and path planning co-optimization (CoMRF-Opt)

$$
\begin{aligned}
\max_{\gamma,\eta,z,u}\quad & (\gamma+\eta)\cdot[r,0]^T \\
\text{s.t.}\quad & (1)\ \sum_{j\in\{A(i)\cup i\}}\gamma_j+\eta_j \le 1, \quad \forall i\in\{1,...,N\}, \\
& (2)\ \sum_{i=1}^{N+1}\sum_{j=1}^{N+1} z_{i,j}d_{i,j} \le \mathscr{E}, \\
& (3)\ \sum_{j=1}^{N+1} z_{i,j} = \sum_{j=1}^{N+1} z_{j,i} = \eta_i,\ \forall i\in\{1,...,N+1\}, \\
& (4)\ u_i-u_j+1 \le N\cdot(1-z_{i,j}),\ \forall i,j\in\{2,...,N+1\}, \\
& (5)\ \mathbf{1}^T\gamma \le M, \qquad (6)\ \gamma+\eta \preceq \mathbf{1}, \\
& (7)\ \gamma_{N+1}=0,\ \eta_{N+1}=1, \\
& (8)\ \gamma,\eta\in\{0,1\}^{N+1},\ z\in\{0,1\}^{(N+1)^2},\ u\in[2,N+1]^N
\end{aligned}
$$

the nodes that maximize uncertainty reduction and avoids choosing highly-correlated nodes, under limited query and motion budgets.[7] We assume that the robot has to return to its initial position after completing the close-up sensings, forming a tour. The optimization formulation is described in Alg. 1, where N is the total number of objects that the robot has initially sensed, $\gamma=[\gamma_1,...,\gamma_{N+1}]$ denotes the binary decisions of querying the nodes, $\eta=[\eta_1,...,\eta_{N+1}]$ denotes the binary decisions of visiting the nodes, γ_{N+1} and η_{N+1} are augmented variables for the robot's initial position, $r=[r_1,...,r_N]$ is the uncertainty vector from Eq. (2), $\mathscr{E}$ is the motion budget in terms of total traveled distance, and M is the query budget. γ_i=1 indicates that the robot will query object i (with 0 denoting otherwise). $\eta_i=1$ indicates that the robot will visit object i (with 0 denoting otherwise).

In Constraint (1), we impose that for each node i, at most one node can be selected from the set of node i and its neighbors, which prevents the simultaneous selection of highly-correlated nodes. Constraints (2)-(4) are related to the robot's tour planning. Constraint (2) limits the total traveled distance by $\mathscr{E}$, where $d_{i,j}$ is the distance between objects i and j, and $z_{i,j}\in\{0,1\}$ indicates whether to include edge (i,j) in the tour. Constraint (3) restricts that an object location can only be entered and exited once if it is in the tour ($\eta_i=1$). Constraint (4) is the Miller-Tucker-Zemlin (MTZ) constraint that eliminates sub-tours [37]. Constraint (5) limits the number of queries by M. Constraint (6) prohibits the robot from both querying an object and visiting it. Constraint (7) ensures that the initial robot position is part of the tour. The last set of constraints enforce that all the decision variables (γ, η, and z) are binary, and that the MTZ variables are in $[2,N+1]$.

In the solution of Alg. 1, it is possible that not all the given query and motion budgets are utilized, since we enforce Constraint (1) to avoid selecting highly-similar nodes. Suppose that Ω is the set of nodes queried or visited in this solution. If there are any unused queries and/or motion budget, the robot then re-runs a slightly-modified version of Alg. 1, where $w=0$, Constraint (1) is removed, and the nodes in Ω are enforced to be queried or visited. In this way, the robot prioritizes the selection of the most important nodes in CoMRF, while ensuring that all the resources are properly utilized. For the rest of the paper, we refer to our proposed query selection and path planning approach as **CoMRF-Opt**.[8]

IV. Performance Evaluation on Large COCO-based Test Set

In this section, we evaluate our proposed CoMRF-based co-optimization approach on joint labeling, query selection, and path planning on the large COCO-based test set described in Sec. II-A. We first assume no query or motion budgets for the robot, and evaluate our proposed joint labeling method. We then allow several queries for the robot (no motion) and evaluate our query selection method. Lastly, we incorporate the element of motion and evaluate our query-motion co-optimization approach by simulating the motion.

A. Joint Labeling

In this part, we assume zero query and motion budgets for the robot, and analyze the joint labeling performance. We compare with the state-of-the-art methods of Cao et al. [19] and Hayder et al. [20], which are similarity-based approaches that use hand-crafted image features to deduce image similarity for joint labeling. We further compare with the benchmark of directly using the base classifier's output, to which we refer as "independent".[9]

For each test case, we randomly draw N_p classes from the total $N_c=39$ classes. For each selected class, we then sample N_I images. In this section, we use $N_p=2$ and $N_I=50$, and report the average classification accuracy over 100 random test cases. To make the classification task more challenging, the images are randomly sampled from the set of images whose AlexNet initial classification confidence is below 0.9.

Table II compares the performance of the independent approach, Cao et al., Hayder et al., and our proposed CoMRF-based joint labeling. We can see that our approach considerably outperforms the independent approach when using any of the DCNN classifiers, with an average improvement of 0.195 in terms of classification accuracy. On the other hand, Cao et al. and Hayder et al. provide only slight improvement in the case of AlexNet, and underperform, as compared to the base classifier, in the cases of the other two DCNNs. This is because their hand-crafted features cannot properly capture object similarity, especially when there is a large number of classes with a wide variety of poses and views for each class.

B. Query Selection

We next evaluate our proposed query selection approach (using Alg. 1 with a zero motion budget). In this evaluation, we compare with Wang et al. [25], which is a state-of-the-art Bayesian approach. Since this approach is for selecting queries on a given MRF and not on image similarity, we use

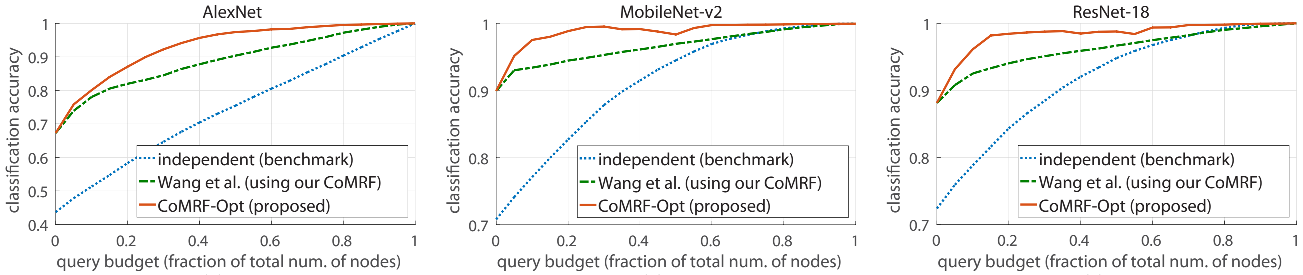
[7] In this optimization formulation, the robot is assumed to obtain the correct label if it visits an object. In the experiments of Sec. VII-C, the reported performance is based on the actual obtained images after the visits.

[8] The optimization in Alg. 1 is a Mixed Integer Linear Program (MILP) and we use Matlab's MILP solver in Secs. IV, V, VI, and VII.

[9] More details of these methods can be found in Appendix B.

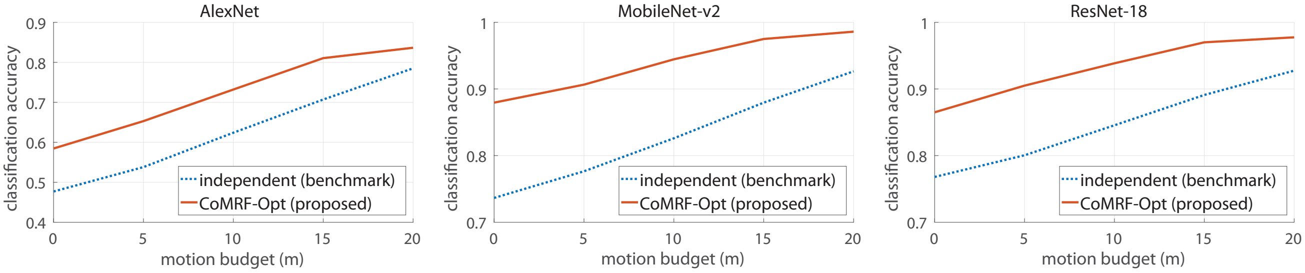
Base classifier	Independent (base classifier output)	Cao et al. [19]	Hayder et al. [20]	CoMRF (proposed)
AlexNet	0.437	0.438	0.543	0.673
MobileNet-v2	0.708	0.422	0.587	0.900
ResNet-18	0.724	0.422	0.604	0.881
average	0.623	0.427	0.578	0.818

TABLE II: Performance comparison of joint labeling methods (case of no query or motion). It can be seen that our proposed joint labeling method improves the base classifier by 0.195, on average, in terms of classification accuracy. On the other hand, Cao et al. and Hayder et al. perform similar or worse as compared to the initial classification.



(a) (b) (c)

Fig. 4: Joint labeling and query selection performance (no motion) on the COCO-based test set. The query budget is given as a fraction of the total number of nodes.



(a) (b) (c)

Fig. 5: Query selection and path planning performance on the COCO-based test set. The robot is given a query budget equal to 5% of the total number of nodes and the motion budget ranges from 0 m to 20 m.

our CoMRF, but then apply their query strategy instead of our proposed Alg. 1. We also include a benchmark that greedily selects the nodes with the highest individual uncertainties (from the base classifier), without utilizing any correlation, to which we refer as "independent".[10]

The test cases used in this part are the same as in Sec. IV-A, and the reported classification accuracy is averaged over the 100 test cases. Fig. 4 compares the performance of our CoMRF-Opt (red solid) with the independent approach (blue dashed) and Wang et al. (green dashed), when using the three base classifiers, respectively. As can be seen, our proposed approach considerably outperforms both of them. For instance, when using AlexNet as the base classifier, given a budget of 40 queries, CoMRF-Opt achieves a classification accuracy of 0.957, as compared to 0.878 of Wang et al. and 0.704 of the independent approach. As for the other two DCNNs, although CoMRF already achieves a high initial classification accuracy of ~ 0.900, CoMRF-Opt is still able to further improve it to ~ 0.990 with 20 queries, significantly outperforming Wang et al. and the independent approach. Furthermore, our proposed approach enables significant resource savings. For instance, for the case of AlexNet, in order to achieve an average classification accuracy of 0.900, our proposed method requires 25 queries, while Wang et al. and the independent benchmark require 50 and 80 queries, respectively, as shown in Fig. 4 (a). This is equivalent to respective reductions of 50% and 68.75% in terms of communication resources.

Since Cao et al. and Hayder et al. only perform comparable to or worse than the base classifier, and Wang et al. only provides near-linear improvement with respect to the number of given queries, we will not include them for comparison in the rest of the paper.

C. Query Selection and Path Planning

In this section, we take motion into account and evaluate our CoMRF-based query-motion co-optimization approach on our COCO-based test set by running the robot in a simulated motion environment. More specifically, for each test case, we randomly draw $N_p = 2$ classes from the total $N_c = 39$ classes and for each selected class, we randomly sample $N_I = 10$ images from the set of images whose AlexNet initial classification confidence is below 0.9. These 20 images/objects are then randomly placed in a $10\,\mathrm{m} \times 10\,\mathrm{m}$ simulation environment. In each test case, given a query budget and a motion budget, the robot needs to decide for which object images it should query the remote human operator and which object locations it should visit to sense better. We give the robot a query budget equal to 5% of the total number of nodes and a motion budget ranging from 0 m to 20 m. We compare our proposed

[10] More details of these methods can be found in Appendix B.

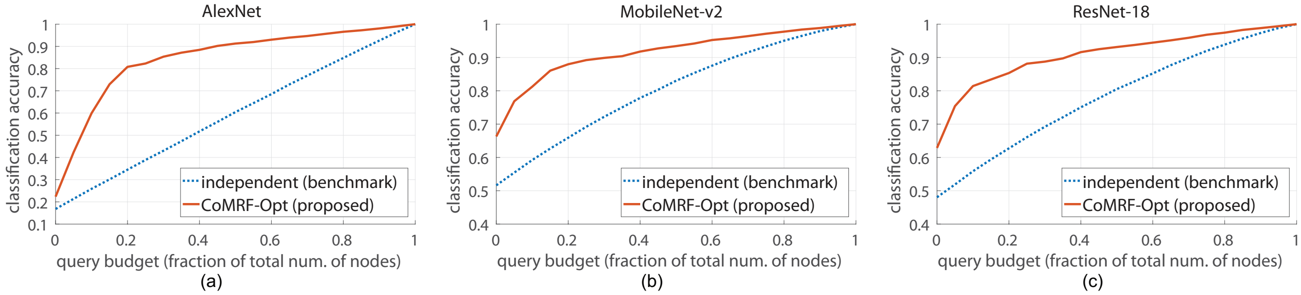


Fig. 6: Joint labeling and query selection performance (no motion) on the VisDrone dataset, for the same-flight scenario. The query budget is given as a fraction of the total number of nodes.

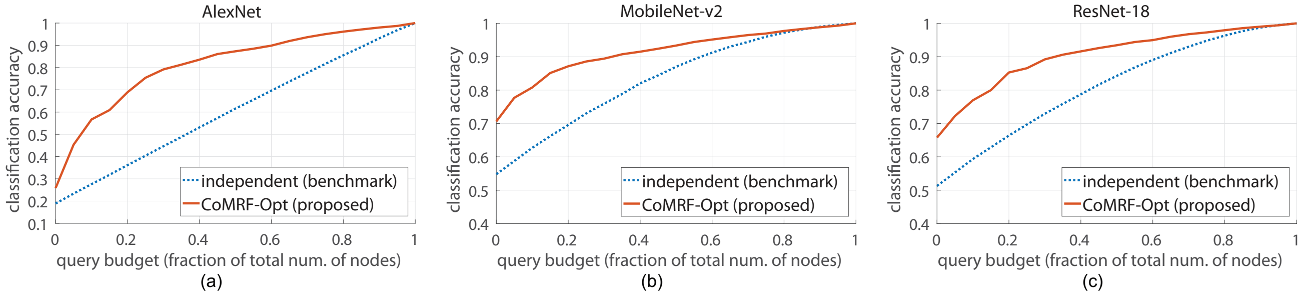


Fig. 7: Joint labeling and query selection performance (no motion) on the VisDrone dataset, for the multi-flight scenario. The query budget is given as a fraction of the total number of nodes.

CoMRF-Opt (Alg. 1) with the independent approach that does not take into account any correlation.[11] We report the average classification accuracy over 100 random test cases.

Fig. 5 shows the classification performance of CoMRF-Opt (red solid) and that of the independent approach (blue dashed), for the cases of the three DCNN base classifiers, respectively. It can be seen that CoMRF-Opt significantly outperforms the independent approach for any given motion budget and for all three DCNNs. For instance, in the case of MobileNet-v2, given a motion budget of 10 m, CoMRF-Opt achieves a classification accuracy of 0.945, which is considerably higher than that of the independent benchmark (0.826). For a larger query budget, our proposed approach has a similar performance improvement over the benchmark. For instance, given a query budget of 15% and a motion budget of 10 m, CoMRF-Opt has a classification accuracy of 0.978 and the benchmark's accuracy is 0.880, when using MobileNet-v2. As the amount of given resources increases considerably, however, both approaches' performance will approach 1 eventually, as expected.

Overall, these results confirm that our proposed feature correlation metric robustly captures image similarity and our CoMRF-based approach improves classification performance considerably. For the joint labeling task, the accuracy of our proposed approach is significantly higher than those of the state of the art. By comparing with Wang et al., we can see that our proposed CoMRF-based query strategy outperforms one of the best existing approaches. By including motion in the experiments, we further validate our proposed CoMRF-based query-motion co-optimization methodology.

V. Performance Evaluation on a Large-Scale Drone Imagery Dataset

In this section, we evaluate the performance of our proposed CoMRF-based joint labeling and query-motion co-optimization methodology on a large drone imagery dataset. We use the publicly-available VisDrone dataset [38],[12] which is a challenging large-scale dataset with images captured by drone-mounted cameras that cover various cities, environments, object classes, and object densities. As images taken by drones have very different views as compared to images taken on the ground (e.g., COCO images), evaluating performance on the VisDrone dataset will further verify the robustness and generalizability of our proposed feature correlation-based similarity metric and the CoMRF-based co-optimization approach. While using the VisDrone data, we consider a subset of their object classes (7 out of 10) that overlaps with the classes of our training set described in Sec. II-A, which includes person, pedestrian, car, bus, truck, motorcycle, and bicycle. The two classes of person and pedestrian are treated as one class in our evaluation. The object image patches are obtained based on the provided bounding box annotations.

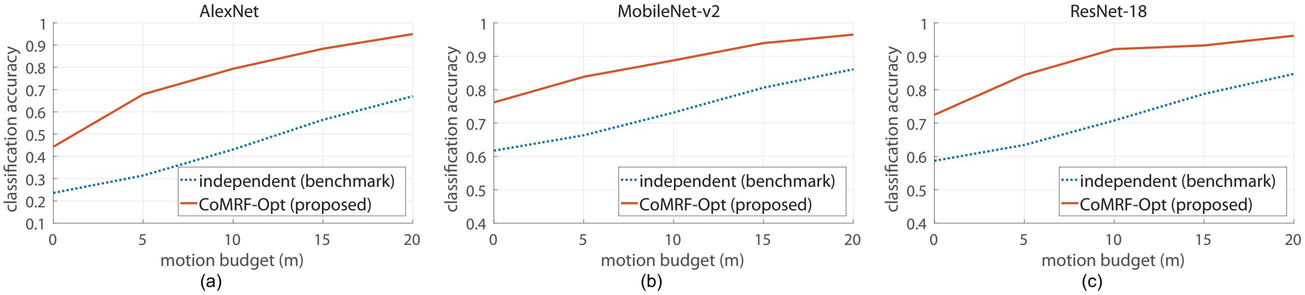
In the following parts of this section, we first evaluate the performance of joint labeling and query selection by assuming a zero motion budget. We then allow a non-zero motion budget and evaluate our proposed query-motion co-optimization approach in a simulation environment.

A. Joint Labeling and Query Selection

In the joint labeling and query selection evaluation, we consider two realistic drone visual sensing scenarios. In the first scenario, in each test case, the images are taken from the same flight. This captures a real-world scenario where the

[11] For the independent approach, the path planning part is conducted by running a modified version of Alg. 1, where r in the objective function is replaced by a vector of the individual uncertainties, given by the base classifier, and Constraint (1) is removed.

[12] The dataset is publicly available from http://aiskyeye.com/.



(a) (b) (c)

Fig. 8: Query selection and path planning performance on the VisDrone dataset. The robot is given a query budget equal to 5% of the total number of nodes and the motion budget ranges from 0 m to 20 m.

N_p	Average N_I	AlexNet		MobileNet-v2		ResNet-18	
		Independent	CoMRF-Opt	Independent	CoMRF-Opt	Independent	CoMRF-Opt
≤ 3	41	0.593	0.862	0.804	0.916	0.777	0.909
4	27	0.602	0.844	0.815	0.917	0.796	0.909
5	22	0.585	0.846	0.797	0.921	0.770	0.918
6	18	0.617	0.798	0.799	0.887	0.787	0.862

TABLE III: Average classification performance given different number of classes present (N_p) in a test case. The first column shows the number of classes present in a test case. The second column shows the average number of objects per class in a test case for different N_p values. The remaining columns show the average classification accuracy of the benchmark of making independent decisions and our proposed CoMRF-based approach, given different N_p values, for the base classifiers of AlexNet, MobileNet-v2, and ResNet-18, respectively.

drone utilizes not only the correlation among objects within the same image, but also that among objects from different images taken during the same flight, in order to improve its onboard DCNN classifier's accuracy. We refer to this scenario as the **same-flight scenario**. Secondly, we consider a scenario where in each test case, the images are randomly drawn from the entire dataset. This captures another practical scenario where there are multiple drones performing visual tasks in different environments, while there is a central agent (e.g., a leader drone or a ground robot) that receives images from these drones and performs joint labeling. We refer to this scenario as the **multi-flight scenario**. In both scenarios, each test case contains at least 100 objects from images from either the same flight or multiple random flights, and we report the average classification accuracy over 100 random test cases for each scenario. Note that, unlike in the previous section, the number of classes present (N_p) and the number of images per class (N_I) are not controlled here. In other words, there can be any number of at least 2 object classes present and the number of images for each class can take any value in each test case.

Fig. 6 shows the performance of our proposed joint labeling and query selection approach, as compared to the benchmark, in the same-flight scenario. It can be seen that our proposed approach significantly outperforms the benchmark in terms of classification accuracy, when using any of the three DCNN base classifiers. For instance, when using AlexNet, given a query budget equal to 30% of the number of nodes, CoMRF-Opt achieves an accuracy of 0.854, which is 0.424 higher than that of the base classifier (0.430). Similar large improvements can be seen for the cases of MobileNet-v2 and ResNet-18.

Fig. 7 compares our proposed CoMRF-based approach with the independent benchmark, in the multi-flight scenario. Similarly, it can be seen that our proposed approach significantly outperforms the independent approach, for all three DCNNs. For instance, in the case of AlexNet, given a query budget equal to 30% of the number of nodes, CoMRF-Opt achieves an accuracy of 0.792, while the benchmark has an accuracy of 0.447, which is 0.345 lower. It can also be observed that the performance improvement provided by CoMRF-Opt is slightly less in the multi-flight scenario, as compared to that in the same-flight scenario, since there could be less correlation across images taken from different flights, as expected.

Next, we study how the classification performance varies w.r.t. the number of classes present (N_p) in a test case, in order to understand the effect of N_p on our proposed approach. More specifically, based on the 100 same-flight test cases, we calculate the average classification accuracy for each N_p, where the accuracy is averaged over the test cases with the same N_p and the query budget ranging from 0 to 1. The results are shown in Table III. The first column shows the different N_p values, where $N_p = 2$ and $N_p = 3$ are grouped together to allow for at least 10 cases for averaging. The second column shows the average number of objects per class (N_I) in a test case, for each N_p. The remaining entries show the average classification accuracies of the benchmark of making independent decisions and our CoMRF-based approach. It can be seen that for each base classifier, the benchmark performs similarly across different N_p values, which is as expected as it classifies each object image individually and is thus not affected by N_p. As for our approach, for each base classifier, we can see that the average classification accuracy is also similar across different N_p values. The accuracy is slightly lower when $N_p = 6$, as the average N_I (number of objects per class) is smaller in this case, which means that there is less underlying correlation to be exploited. However, as we can see, our CoMRF-based approach still significantly outperforms the benchmark in this case. These results verify that our proposed approach performs consistently across different N_p values.

B. Query Selection and Path Planning

In this part, we provide a non-zero motion budget to the robot and evaluate our proposed query-motion co-optimization

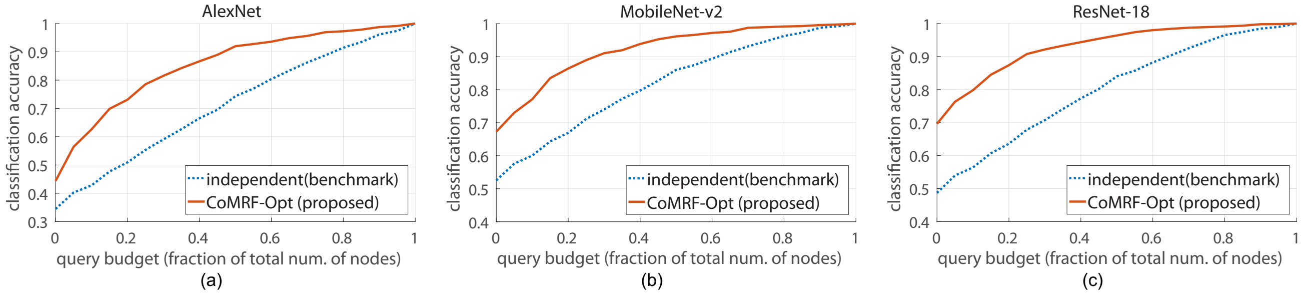


Fig. 9: Joint labeling and query selection performance (no motion) on the NYU-v2 dataset. The query budget is given as a fraction of the total number of nodes.

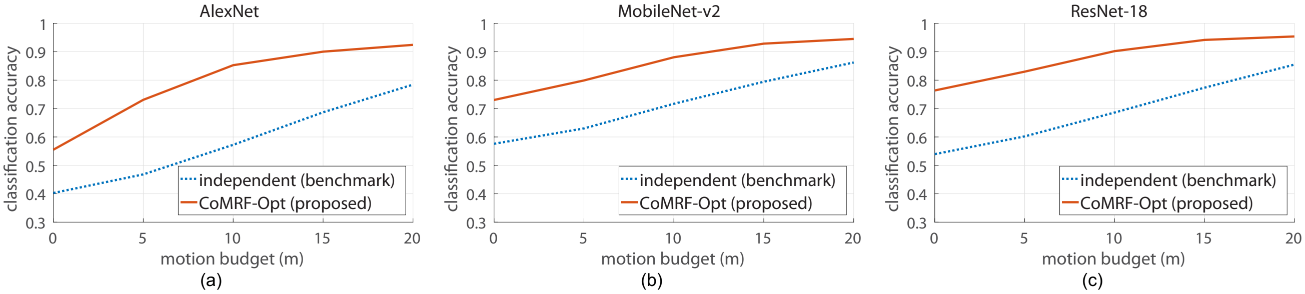


Fig. 10: Query selection and path planning performance on the NYU-v2 dataset. The robot is given a query budget equal to 5% of the total number of nodes and the motion budget ranges from 0 m to 20 m.

approach. In each test case, an image (with 15 to 25 objects) is randomly drawn from the VisDrone dataset and the objects in this image are randomly placed in a 10 m × 10 m simulation environment. By running our proposed algorithm in this setting, we capture a realistic scenario where a drone has acquired an image of the field, and needs to plan its next motion steps to better view the objects and/or select some of the object images to query the remote human operator. We average the performance over 100 random test cases.

Fig. 8 shows the performance of our proposed CoMRF-Opt query-motion co-optimization and that of the independent approach, for the three DCNN base classifiers, respectively. It can be seen that overall, CoMRF-Opt significantly outperforms the independent approach. For instance, when using ResNet-18, given a motion budget of 10 m, CoMRF-Opt achieves a classification accuracy of 0.921, as compared to an accuracy of 0.707 by the independent benchmark.

Overall, these results confirm the efficacy of our proposed CoMRF-based joint labeling and query-motion co-optimization approach, showing that CoMRF-Opt can achieve significantly higher classification accuracies as compared to the benchmark of making independent decisions. Furthermore, the visually-challenging VisDrone-based evaluation also demonstrates the robustness and generalizability of our proposed feature correlation and co-optimization approach.

VI. Performance Evaluation on a Large Indoor Scene Dataset

We further evaluate the performance of our proposed CoMRF-based joint labeling and co-optimization methodology on the popular indoor scene dataset of NYU-v2 [39].[13] This large indoor dataset contains a variety of scenes (e.g., kitchens, offices) and a large number of various objects. As indoor objects can be challenging to recognize, e.g., due to occlusion and clutter, this evaluation will further verify the robustness of our proposed approach. In the evaluation, we consider a set of 19 object classes that are present in both the NYU-v2 dataset and our training set in Sec. II-A. The object image patches are obtained based on the provided annotations.

In the evaluation, we use the objects of the same scene for each test case, which captures a real-world situation where a robot enters a scene and needs to recognize the objects in the scene. We randomly select 100 test cases (i.e., 100 scenes) from the dataset, each contains at least 10 objects. The reported performance is averaged over the 100 test cases. Next, we present the evaluation on joint labeling, query selection, and query-motion co-optimization for our proposed approach.

A. Joint Labeling and Query Selection

Fig. 9 shows the performance of our proposed joint labeling and query selection approach, as compared to the benchmark of making independent decisions. It can be seen that our proposed approach significantly outperforms the independent approach, when using any of the three DCNN base classifiers. For instance, given a query budget equal to 30% of the number of nodes, CoMRF-Opt achieves an accuracy of 0.815, 0.911, and 0.922, considerably outperforming the benchmark by 0.225, 0.170, and 0.214, respectively, when using the base classifiers of AlexNet, MobileNet-v2, and ResNet-18.

B. Query Selection and Path Planning

In this part, the robot is given a non-zero motion budget. In each test case, the objects from the same scene are randomly placed in a 10 m × 10 m simulation environment. Our evaluation in this section captures a real-world scenario where a robot is tasked with recognizing the objects in an indoor

[13] The dataset is publicly available from https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html.

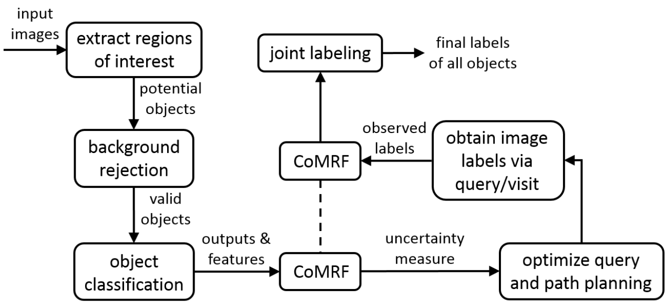


Fig. 11: The diagram shows the robot's steps in an experiment using our proposed approach. The dashed line between the two "CoMRF" blocks indicates that they are the same MRF, based on the initial classification and feature extraction. The top CoMRF is then further updated with the image labels obtained from visit or query.

scene, and needs to plan its motion steps to better view the objects and/or select some of the object images to query the remote human operator.

Fig. 10 shows the performance of our proposed CoMRF-Opt query-motion co-optimization approach and that of the independent one, for the three DCNN base classifiers. It can be seen that CoMRF-Opt significantly outperforms the independent approach. For instance, when using AlexNet, given a motion budget of 10 m, CoMRF-Opt achieves a classification accuracy of 0.853, as compared to an accuracy of 0.572 by the independent benchmark.

Overall, these results further confirm the efficacy and robustness of our proposed CoMRF-based joint labeling and query-motion co-optimization approach for indoor scenes, showing that it can achieve a significantly higher classification accuracy as compared to making independent decisions.

VII. Robotic Experiments

In this section, we evaluate our proposed query selection and path planning approach (CoMRF-Opt) with several robotic experiments on our campus. We also compare its performance with the benchmark of making independent decisions. We have conducted a total of six experiments. In the first three experiments, we evaluate the query selection part, where the robot takes several images around it and is allowed to ask for human help under a query budget (zero motion budget). In the remaining three experiments, we provide the robot with a motion budget, adding path planning into the robot's decision-making. The real-world robotic experiments are conducted with the robot running the AlexNet classifier onboard.

A. Experiment Overview

In the experiments, we use a Pioneer 3-AT robot, equipped with a ZED camera (with depth sensing) and a laptop. Fig. 11 shows the robot's steps during an experiment. The robot first takes several images of its surroundings. For each image, the robot extracts Regions-Of-Interest (ROIs), each of which potentially contains an object-of-interest.[14] These potential objects are then fed into a background rejector, which is a binary classifier that determines whether an image contains an object-of-interest (e.g., one of the 39 object classes vs. a background wall).[15] Once the background image patches are rejected, the robot passes the object image patches, each containing an object-of-interest, to its onboard DCNN classifier (described in Sec. II-A), which provides the classification output and feature vector for each object image. The robot then constructs the CoMRF for these objects, and optimizes (in real time) the queries and motion using Alg. 1. Given the optimized decisions, the robot performs the corresponding queries and/or further sensing, after which it obtains the labels for the queried/visited nodes and updates the remaining nodes on the CoMRF, using LBP as discussed in Sec. II-B.

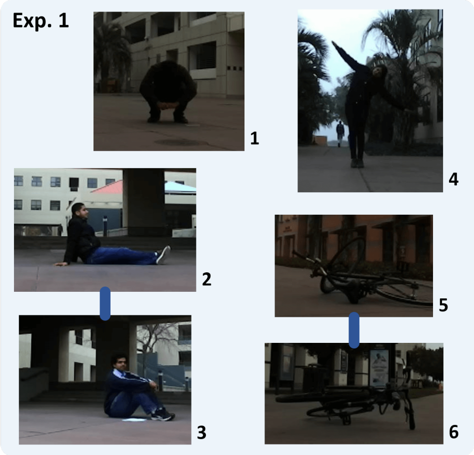


Fig. 12: Object images obtained by the robot in Exp. 1 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF. See the color pdf to better view all the experiment images.

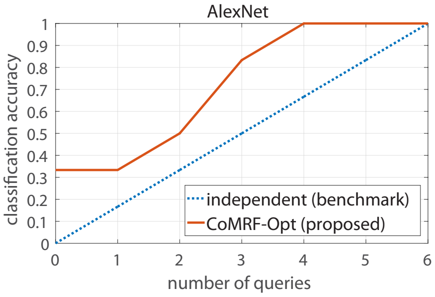


Fig. 13: Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries in Exp. 1 (shown in Fig. 12).

B. Query Selection

In the first three experiments, the robot is tasked with object classification on our campus, and is allowed to query a remote human operator for help. We assume that when the robot queries about an object, it receives the correct label.

1) Experiment 1: Fig. 12 shows the images taken by the robot in this experiment on our campus. The CoMRF is constructed based on the extracted features and the chosen correlation threshold discussed in Sec. II. In this CoMRF, there

[14] The ROI extraction algorithm utilizes the depth and the saliency map of [40]. This algorithm is designed to find coarse ROIs, each of which may contain an object-of-interest, rather than providing tight bounding boxes.

[15] Details of the ROI extraction algorithm and the background rejection classifier can be found in Appendices C and D.

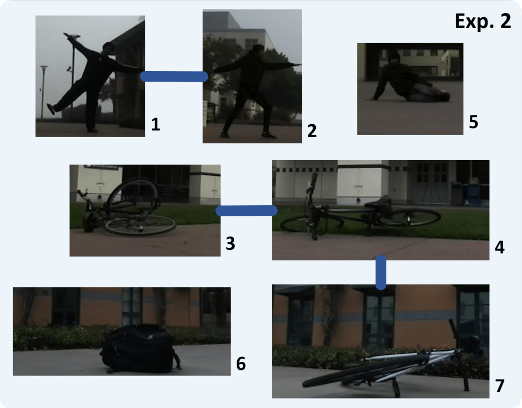


Fig. 14: Object images obtained by the robot in Exp. 2 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

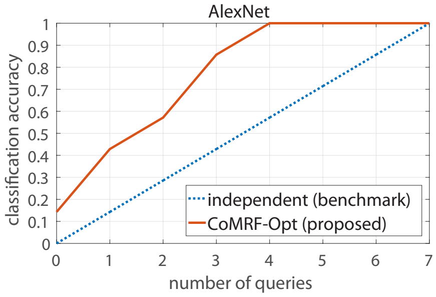


Fig. 15: Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries in Exp. 2 (shown in Fig. 14).

is an edge between images 2 and 3, and an edge between images 5 and 6, as can be seen.[16]

The robot then performs joint labeling and query selection using this CoMRF, and we compare its performance with the independent approach. Fig. 13 shows the two methods' accuracies with respect to the number of queries. Initially, the classifier mislabels all six images. Although the AlexNet classifier has a good accuracy of 0.800 over the validation set, its performance degrades in real-world scenarios due to low resolution and non-ideal lighting. When there are no allowed queries, CoMRF improves the initial classification by correctly labeling images 2 and 3 to persons (using feature correlation and joint labeling). Given some queries, we can see that CoMRF-Opt outperforms the independent benchmark significantly. For instance, when given 4 queries, CoMRF-Opt chooses images 1, 3, 4, and 6, and achieves a 1.000 accuracy. On the other hand, the independent approach chooses both nodes 2 and 3 among the 4 queries, which is unnecessary as they are highly similar, and achieves a 0.667 accuracy.

2) Experiment 2: Fig. 14 shows the images taken by the robot in this campus experiment. In this CoMRF, there is an edge between images 1 and 2, an edge between images 3 and 4, and an edge between images 4 and 7, as can be seen. In particular, the three bicycle images (3, 4, and 7) form a connected component in the graph.

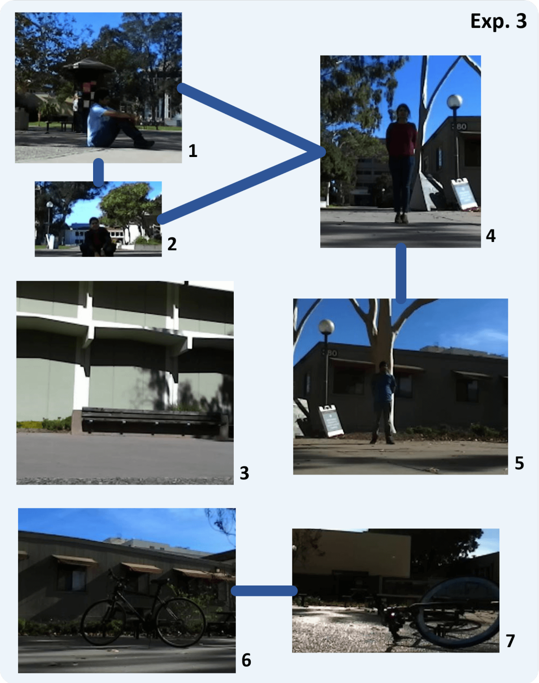


Fig. 16: Object images obtained by the robot in Exp. 3 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

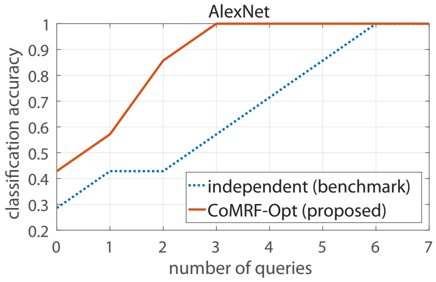


Fig. 17: Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries in Exp. 3 (shown in Fig. 16).

When there are no queries, CoMRF improves the initial classification by correctly labeling node 3 as a bicycle. Initially, nodes 3 and 4 are mislabeled as a potted plant and a bench, respectively, which are the most likely candidates from the classifier for these two nodes, while bicycle is the second most likely for both nodes. After the similarity between nodes 3 and 4 has been captured in CoMRF, the probability of belonging to the bicycle class increases for both nodes, and for node 3, bicycle becomes the most probable class.

When the robot is given several queries, CoMRF-Opt outperforms the independent approach significantly. For instance, when given 4 queries, CoMRF-Opt chooses nodes 2, 4, 5, and 6, and achieves a 1.000 accuracy. The independent approach, on the other hand, chooses both nodes 1 and 2 among the 4 queries, which are highly correlated, only achieving a 0.571 accuracy.

3) Experiment 3: Fig. 16 shows the images taken by the robot in this campus experiment. In this CoMRF, the persons' images form a connected component, with images 1, 2, and 4

[16] As discussed in Sec. II-A, the robot may not capture exhaustively all the pairwise same-class objects as we set the threshold high to make the probability of false correlation very small. But as these experiments indicate, what it captures can lead to significant performance improvements for free, by utilizing our proposed similarity metric.

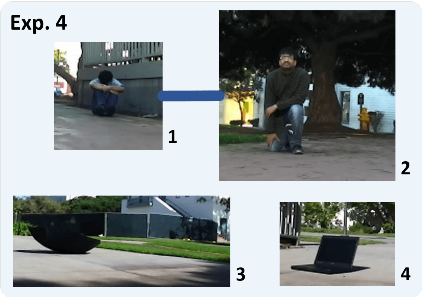


Fig. 18: Object images obtained by the robot in Exp. 4 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

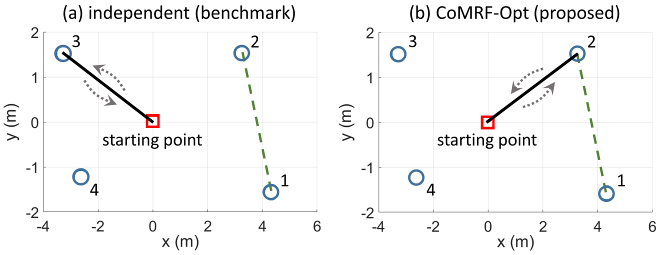


Fig. 19: Blue circles indicate the object locations and the red square indicates the robot's initial position in Exp. 4 of Fig. 18. The green dashed line indicates an edge in the CoMRF. (a)-(b) show the respective path planning results of the independent approach and CoMRF-Opt, where the solid black line indicates the sensing tour.

fully interconnected and image 5 connected to image 4. There is also an edge between the bicycle images 6 and 7.

Initially, the based classifier of AlexNet correctly recognizes images 1 and 4, but misclassifies the rest. By applying CoMRF, the robot is then able to classify image 2 as a person, thus improving the classification accuracy without using any queries. When the robot is given a few chances to query, CoMRF-Opt outperforms the independent approach significantly. For instance, when given 3 queries, CoMRF-Opt chooses nodes 3, 5, and 7, and achieves a 100% classification accuracy. On the other hand, the independent approach chooses nodes 4, 6, and 7, among which nodes 6 and 7 are highly correlated, and only achieves a 0.571 accuracy.

C. Query Selection and Path Planning

In this part, we present robotic experiments where the robot is given a non-zero motion budget. When the robot visits an object, it moves towards the object and takes a close-up image. For an object visited by the robot, the label is given by the actual DCNN classification output based on the close-up image obtained during the visit.[17] The object locations are estimated based on the depth information.

1) Experiment 4: Fig. 18 shows the objects initially captured by the robot in this experiment, all of which are misclassified initially by the base classifier. As can be seen, CoMRF puts an edge between images 1 and 2. Fig. 19 shows the object locations and the robot's initial position. The green dashed line indicates an edge between objects 1 and 2 in the CoMRF. In this experiment, the robot is given no queries and is allowed a total travel distance of 8 m. Fig. 19 (a) and (b) show the results of the independent benchmark and our approach, respectively. The independent approach does not take advantage of the feature correlation and chooses to visit the standalone node 3. On the other hand, our proposed CoMRF-based path planning approach of Sec. III is aware of the correlation and chooses to visit node 2. After node 2 is better sensed and correctly labeled as a person, CoMRF propagates this information to node 1 (via joint labeling) and then also correctly classifies node 1. Therefore, under the same motion budget, CoMRF-Opt correctly classifies one more node as compared to the benchmark and improves the accuracy by 100% over the benchmark.

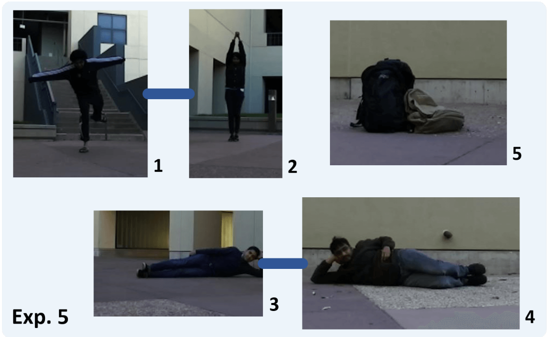


Fig. 20: Object images obtained by the robot in Exp. 5 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

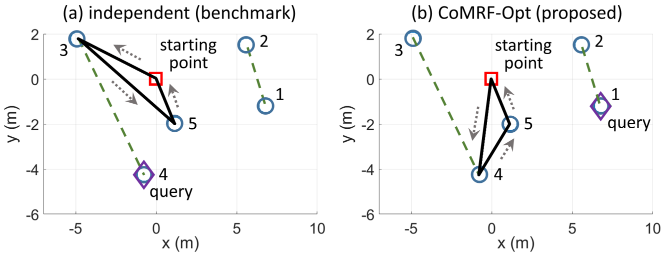


Fig. 21: Blue circles indicate the object locations and the red square indicates the robot's initial position in Exp. 5 of Fig. 20. The green dashed line indicates an edge in the CoMRF. (a)-(b) show the respective path planning results of the independent approach and CoMRF-Opt, where the solid black line indicates the sensing tour and the purple diamond indicates the queried object.

2) Experiment 5: Fig. 20 shows the object images captured by the robot in this experiment. In the CoMRF, there is an edge between images 1 and 2, and an edge between images 3 and 4. Fig. 21 shows the object locations and the robot's initial position. In this case, the robot is given 1 query and is allowed a total travel distance of 16 m. Fig. 21 (a) shows the result of the independent approach. As can be seen, the robot visits nodes 3 and 5, and queries node 4. However, since nodes 3 and 4 are highly correlated, it is unnecessary to obtain labels for both of them. On the other hand, as shown in Fig. 21 (b), CoMRF-Opt queries node 1, and visits nodes 4 and 5, which are not highly correlated. Furthermore, due to their influence on their neighboring nodes, CoMRF correctly labels the remaining nodes after the querying and sensing. In this case, the independent approach only obtains a classification accuracy of 0.600, while CoMRF-Opt provides

[17] For the independent approach, we assume that the robot obtains the correct label of the visited node when calculating its performance.

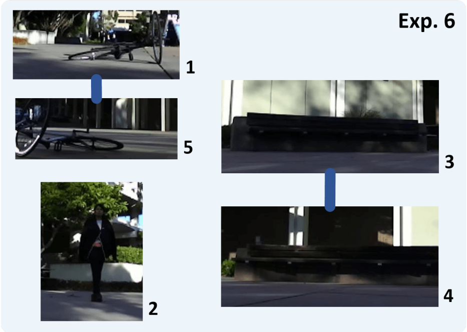


Fig. 22: Object images obtained by the robot in Exp. 6 on our campus. A line between two images indicates that there is an edge between them in the corresponding CoMRF.

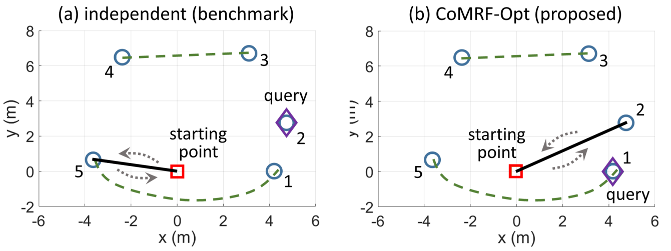


Fig. 23: Blue circles indicate the object locations and the red square indicates the robot's initial position in Exp. 6 of Fig. 22. The green dashed line indicates an edge in the CoMRF. (a)-(b) show the respective path planning results of the independent approach and CoMRF-Opt, where the solid black line indicates the sensing tour and the purple diamond indicates the queried object.

fully correct classifications, significantly improving the accuracy of the benchmark by 0.400.

3) Experiment 6: Fig. 22 shows the object images captured by the robot in Exp. 6. In the CoMRF, there is an edge between images 1 and 5, and an edge between images 3 and 4. Fig. 23 shows the object locations and the robot's initial position. In this case, the robot is given 1 query and is allowed a total travel distance of 11 m. Fig. 23 (a) and (b) shows the result of the independent approach and CoMRF-Opt, respectively. Initially, the robot correctly recognizes images 3 and 4 as benches. By using our proposed approach, the robot visits node 2 and queries node 1, after which it is able to propagate the newly-acquired information to node 5 and correctly classifies all the images. On the other hand, the independent approach correctly classifies node 2 and 5 via query/visit, but cannot rectify its initial misclassification of node 1. In this case, the independent approach obtains a final classification accuracy of 0.800, while CoMRF-Opt has a perfect accuracy, considerably outperforming the benchmark by 0.200.

D. Joint Classification of Objects from Multiple Scenes

In this part, we consider a joint classification scenario where the robot is required to classify the object images that it has acquired during earlier visits of several scenes. This setting captures the case where the robot is not required to produce the classification results on the spot and thus allows the robot to discover a richer correlation structure from a larger image pool. In this evaluation, all the 34 object images obtained from our experiments are jointly classified. Although some of the people appear more than once in this pool, they appear in very different conditions, e.g., clothing, poses, views, lighting, contrast, and scenes. As such, two images of the same person taken at different times/locations cannot be trivially declared to be connected in the graph and their similarity has to be determined by their corresponding DCNN features.

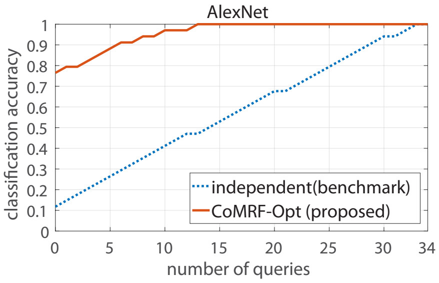


Fig. 24: Performance (classification accuracy) of CoMRF and the independent approach w.r.t. the number of allowed queries, when jointly classifying all the experimental images (shown in Figs. 12, 14, 16, 18, 20, and 22).

Fig. 24 shows the respective classification accuracies of our proposed approach and the independent benchmark in this case. It can be seen that by taking into account the similarity information of all the images, CoMRF is able to greatly improve the classification accuracy without using any queries, from 0.112 to 0.765. By using only 14 queries, CoMRF-Opt achieves a 100% accuracy, while the independent approach requires 33 queries to correctly classify all the images.

Overall, our robotic experiments confirm that the correlation coefficient of two feature vectors, from a DCNN classifier, provides key information on object similarity, and that our proposed CoMRF-based query-motion co-optimization considerably improves the robot's classification accuracy, as compared to the benchmark of making independent decisions.

VIII. Discussion

In this section, we discuss a few more aspects related to our proposed methodology.

A. CoMRF on Other DCNNs

We have shown extensive evaluation of our proposed CoMRF-based approach using the AlexNet, MobileNet-v2, and ResNet-18 DCNN architectures. These are the commonly-used state-of-the-art architectures which are typically suitable for mobile computing (e.g., service robots, drones).

There are even deeper architectures that can be used for classification, at the cost of higher computation and memory requirements. Our proposed correlation-based image similarity and joint labeling are also applicable to such larger and deeper networks, such as Inception-v3 [4] and ResNet-101 [6]. To illustrate this, we have trained these two networks using the training set of Sec. II-A. As shown in Fig. 25, for these two DCNNs, there is a large separation between the distributions of the feature correlation of different-class and same-class pairs. We then run the same joint labeling evaluation, as in Table II of Sec. IV, using these deeper base classifiers. By using Inception-v3, our proposed approach achieves a classification accuracy of 0.937, as compared to the base

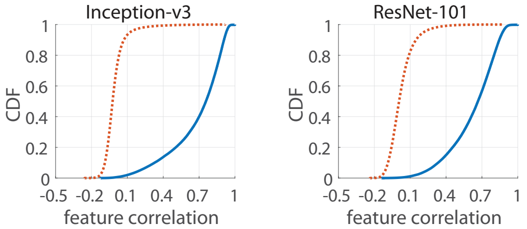


Fig. 25: Each figure shows the CDFs of the feature correlation of same-class (blue solid) and different-class (red dashed) pairs, based on the Inception-v3 and ResNet-101 classifiers, respectively.

classifier's accuracy of 0.814, and by using ResNet-101, our approach achieves a high accuracy of 0.959, as compared to the initial accuracy of 0.818. This demonstrates that our proposed CoMRF is also able to provide a large accuracy improvement with deeper DCNN classifiers.

Furthermore, the performance improvement provided by our proposed query/motion optimization approach also applies to these deeper DCNNs. For instance, when we run the same-flight query selection evaluation on the VisDrone dataset (as described in Sec. V-A), using these two deeper classifiers, with a query budget equal to 10% of the number of objects, our proposed CoMRF-Opt improves the accuracy of the independent benchmark from 0.636 to 0.793 in the case of Inception-v3 and from 0.689 to 0.844 in the case of ResNet-101.

B. Computation Time

In this part, we discuss the computation efficiency of CoMRF-Opt. The timing experiments were run in Matlab on a 3.40 GHz Intel Core i7 PC. The reported times are averaged over 100 problem instances. For a test case with 100 object images/nodes, our proposed query selection algorithm (zero motion budget) only takes an average of 0.012 s to produce a solution. As for the query-motion co-optimization (non-zero query and motion), it takes an average of 0.101 s, 1.363 s, 20.217 s, and 101.585 s to solve a problem with 10, 20, 30, and 40 object images, respectively.

In the path planning part of our algorithm, the number of binary variables increases quadratically with respect to the number of objects, as the robot needs to decide whether to include an edge between every pair of object locations in its trip (see variable $z_{i,j}$ in Alg. 1). However, in a practical mobile robotic visual sensing scenario, it is not very likely that the robot would need to solve the optimization problem with a very large number of sensing locations. For instance, some of the objects may be near each other and can thus share one sensing location. As such, the robot can group nearby objects and solve the planning problem with fewer locations, when the total number of objects in the scene is large.

C. Detection Networks

In this paper, we did not use end-to-end detection architectures (e.g., Faster-RCNN [1]). This is because such detection networks tend to miss a lot of objects, especially in practical robotic settings where the visual recognition can be difficult due to non-ideal lighting, low resolution, small object size, and uncommon viewpoints, as we have observed in the early stage of this study. In fact, on the COCO detection leaderboard, the best method has an average recall of 0.727 for medium-sized objects, indicating that it can miss many objects. In addition, the commonly-used Faster-RCNN with ResNet-101 pipeline only has an average recall of 0.553.[18] Using such detection models makes it difficult for the robot to improve its recognition performance, as it would not even discover several objects in the first place. Thus, in our robotic experiments, we utilize a saliency and depth-based method to discover potential objects near the robot, independent of the recognition difficulty for the onboard DCNN. This ROI extraction method works effectively for our campus experiments even though there are many visually hard-to-detect objects. Its performance, however, may degrade when there is a lot of visual clutter and/or the target objects are not salient, which may result in inaccurate localization or missing objects. Therefore, as part of future work, one could develop a localization/detection model that can discover visually-challenging targets across different scenarios, and integrate our CoMRF-based approach with it.

IX. Conclusions

In this paper, we introduced a training-free object similarity measure, which is based on the correlation of feature vectors provided by a DCNN classifier, to improve robotic visual classification under limited resources. We first probabilistically analyzed the correlation coefficient of the feature vectors of a pair of images from an already-trained DCNN classifier, showing that it provides robust information on their similarity, without requiring any additional training. Based on this analysis, we built a correlation-based Markov Random Field (CoMRF) for joint object labeling. Given a query budget and a motion budget, we then proposed a query-motion co-optimization framework to jointly optimize the robot's query, path, and visual labeling, based on our CoMRF. By using a large COCO-based test set, a large-scale drone imagery dataset, and a large indoor scene dataset, our extensive evaluations showed that our proposed object similarity metric and the resulting CoMRF-based joint labeling and co-optimization methodology significantly improves the overall classification performance. Our several real-world robotic experiments on our campus further showcased the superior performance of our proposed CoMRF-based query-motion co-optimization approach.

References

[1] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, 2015.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[18] Recall is the fraction of the number of correctly identified/classified objects of a certain class over the total number of objects of that class, in an image. More details can be found at http://cocodataset.org/#detection-eval. The average recall of other detection models can be found at http://cocodataset.org/#detection-leaderboard.

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] C. Galleguillos, A. Rabinovich, and S. Belongie. Object categorization using co-occurrence, location and appearance. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[8] A. Torralba, K. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *International Conference on Computer Vision*, 2003.

[9] M. Hasan and A. K. Roy-Chowdhury. Context aware active learning of activity recognition models. In *IEEE International Conference on Computer Vision*, 2015.

[10] A. Jain, A. Zamir, S. Savarese, and A. Saxena. Structural-RNN: Deep learning on spatio-temporal graphs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[12] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics*, 34(4):98, 2015.

[13] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In *ACM International Conference on Multimedia*, 2014.

[14] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[15] E. Prashnani, H. Cai, Y. Mostofi, and P. Sen. PieAPP: Perceptual image-error assessment through pairwise preference. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[16] J. Guérin, O. Gibaru, S. Thiery, and E. Nyiri. CNN features are also great at unsupervised classification. *arXiv:1707.01700*, 2017.

[17] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Robotics: Science and Systems*, 2015.

[18] S. Y. Bao, Y. Xiang, and S. Savarese. Object co-detection. In *European Conference on Computer Vision*, 2012.

[19] L. Cao, J. Luo, and T. S. Huang. Annotating photo collections by label propagation according to multiple similarity cues. In *ACM International Conference on Multimedia*, 2008.

[20] Z. Hayder, M. Salzmann, and X. He. Object co-detection via efficient inference in a fully-connected CRF. In *European Conference on Computer Vision*, 2014.

[21] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena. Contextually guided semantic labeling and search for three-dimensional point clouds. *The International Journal of Robotics Research*, 32(1):19–34, 2013.

[22] H. Ali, F. Shafait, E. Giannakidou, A. Vakali, N. Figueroa, T. Varvadoukas, and N. Mavridis. Contextual object category recognition for RGB-D scene labeling. *Robotics and Autonomous Systems*, 62(2):241–256, 2014.

[23] J. Ruiz-Sarmiento, C. Galindo, and J. González-Jiménez. Joint categorization of objects and rooms for mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015.

[24] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[25] D. Wang, J. Fisher III, and Q. Liu. Efficient observation selection in probabilistic graphical models using Bayesian lower bounds. In *Conference on Uncertainty in Artificial Intelligence*, 2016.

[26] B. Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2012.

[27] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. Guibas. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics*, 35(6):210, 2016.

[28] S. Chen, Y. Li, and N. M. Kwok. Active vision in robotic systems: A survey of recent developments. *International Journal of Robotics Research*, 30(11):1343–1377, 2011.

[29] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42(2):177–196, 2018.

[30] T. Patten, W. Martens, and R. Fitch. Monte Carlo planning for active object classification. *Autonomous Robots*, 42(2):391–421, 2018.

[31] T. Lin et al. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*. 2014.

[32] ImageNet. http://image-net.org/explore.

[33] C. Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[34] D. Koller and N. Friedman. *Probabilistic graphical models: Principles and techniques*. MIT Press, 2009.

[35] UGM: A Matlab toolbox for probabilistic undirected graphical models. http://www.cs.ubc.ca/~schmidtm/Software/UGM.html.

[36] A. Krause and C. Guestrin. Optimal value of information in graphical models. *Journal of Artificial Intelligence Research*, 35:557–591, 2009.

[37] G. Gutin and A. P. Punnen. *The Traveling Salesman Problem and its variations*, volume 12. Springer Science & Business Media, 2006.

[38] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu. Vision meets drones: A challenge. *arXiv:1804.07437*, 2018.

[39] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *European Conference on Computer Vision*, 2012.

[40] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li. Salient object detection: A discriminative regional feature integration approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

Appendix A
Constructed Image Classification Dataset

In this dataset, there is a total of 39 object classes consisting of a variety of commonly-seen objects, and a total of 76,505 images, which are collected from COCO detection dataset [31] and ImageNet [32]. Most of the images are obtained from the COCO detection dataset by extracting object image patches based on the bounding box annotations, in order to better represent what the robot would see in real-world visual tasks. The complete list of object classes, and the numbers of images in the training, validation, and test sets are given in Table IV.

Appendix B
Descriptions of Existing Methods Included in the Performance Comparions

We provide detailed descriptions for the existing methods included in the performance comparisons in Sec. IV.

Joing Labeling

Independent: This is a benchmark method that directly uses the trained DCNN base classifier's output, without considering any correlation.

Cao et al. [19]: This method learns image similarity using hand-crafted features (e.g., SIFT) via a Bayesian approach, after which a propagation algorithm jointly labels all the nodes. Since their algorithm requires an initial set of correct labels, we provide it with 20% of the ground-truth labels in the comparison of Table II.

Hayder et al. [20]: This method uses a Conditional Random Field (CRF) to jointly label nodes, where the edge potential is given by a similarity measure learned from data using hand-crafted features (e.g., Local Binary Pattern) and the node potential is based on the classifier output. To compare this method on our test set, we train their similarity measure on our dataset for the edge potential of their CRF and use our DCNN base classifier's output for the node potential.

Query Selection

Independent: This is a benchmark method that selects the nodes greedily based on their respective individual uncertainty (based on the base classifier's output).

Object class	Training	Validation	Test
person	1000	500	500
bicycle	1000	500	500
car	1000	500	500
motorcycle	905	400	300
airplane	1000	500	500
bus	1000	500	300
train	1000	500	500
truck	1000	500	500
boat	1000	500	500
bench	1000	500	500
bird	1000	500	500
cat	1000	500	500
dog	1000	500	500
horse	1000	500	500
sheep	1000	500	500
cow	1000	500	500
elephant	1000	500	500
zebra	1000	500	500
giraffe	1000	500	500
backpack	950	500	400
umbrella	700	450	300
suitcase	1000	500	300
bottle	1000	500	500
cup	1000	500	500
banana	1000	500	500
apple	1000	500	500
sandwich	1000	500	500
orange	1000	500	500
broccoli	1000	500	500
carrot	1000	500	500
pizza	1000	500	500
donut	1000	500	500
cake	1000	500	500
chair	1000	500	500
potted plant	1000	500	500
laptop	1000	500	500
book	1000	500	500
clock	1000	500	500
teddy bear	1000	500	500
total	38555	19350	18600

TABLE IV: List of object classes in our dataset, along with the numbers of images in the training/validation/test sets.

Wang et al. [25]: Given an undirected probabilistic graphical model (e.g., MRF), this approach selects the nodes to query such that a lower bound of the expected label estimation error of the remaining nodes is minimized.

Appendix C
Region-Of-Interest (ROI) Extraction Algorithm

In the robotic experiments of Sec. VII, we used a simple saliency and depth-based algorithm to find coarse ROIs from the captured images, each of which may contain an object-of-interest. The algorithm is summarized in Alg. 2.

Appendix D
Background Rejection Classifier

Given an image patch from the ROI extraction algorithm, in order to determine whether this image patch does contain an object-of-interest (e.g., an object that belongs to one of the 39 classes vs. a background wall), we train a binary classifier using the AlexNet architecture.

We build the training set for this binary classifier as follows. For objects-of-interest, we use all the training images from our large image classification dataset. For background image patches, we randomly sample image patches that do not overlap with any objects-of-interest from the COCO dataset. A validation set is built similarly. The trained binary classifier has an accuracy of 0.950 on the validation set for distinguishing between object-of-interest images and background images.

Algorithm 2: Region-of-interest extraction

```
INPUT: Image $I$ and its depth map $D$ with $N_w$ and $N_h$ being its width and height (in number of pixels), an upper bound for depth values $d_{\max}$.
STEP 1: Compute a saliency map $S$ for $I$, using the method of [40].
STEP 2: Let $P_d = \{(i,j) \,|\, D(i,j) \in (0, d_{\max})\}$, which is the set of pixel indices with valid depth values.
STEP 3: Let $P_s = \{(i,j) \,|\, S_{i,j} \geq S_{\text{th}}, (i,j) \in P_d\}$, where $S_{\text{th}}$ is a saliency threshold. In our implementation, we set it to be the 75$^{\text{th}}$ percentile of the saliency values of the pixels in $P_d$.
STEP 4: Let $I_{ROI}$ be a ROI indicator map for $I$. We set $I_{ROI}(i,j) = 1$ if$(i,j) \in P_s$ and 0 otherwise. $I_{ROI}$ is then dilated with a $10 \times 10$ kernel.
STEP 5: Find all the connected components (of positive-valued pixels) in $I_{ROI}$. A tight bounding box is drawn around each connected component, and then expanded by increasing its height and width in proportion to the dimensions of the enclosed connected component. The box's height and width are then further expanded by 60 pixels.
STEP 6: Among all the boxes, two boxes are merged under any of the following two conditions: 1) the ratio between their intersected area over the union of their areas is above 0.2, or 2) the ratio between their intersected area over the minimum of their areas is above 0.9. The boxes after the merging are the final ROIs.
```

Hong Cai received the B.E. degree in electronic and and computer engineering from the Hong Kong University of Science and Technology, Kowloon, Hong Kong, in 2013, and the M.S. degree in electrical and computer engineering from the University of California, Santa Barbara, CA, USA, in 2015, where he is currently pursuing the Ph.D. degree in electrical and computer engineering, in the area of communications, control, and signal processing. His research interests include robotic visual understanding and robot decision optimization.

Yasamin Mostofi received the B.S. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 1997, and the M.S. and Ph.D. degrees in the area of wireless communications from Stanford University, California, in 1999 and 2004, respectively. She is currently a professor in the Department of Electrical and Computer Engineering at the University of California Santa Barbara. Dr. Mostofi is the recipient of 2016 Antonio Ruberti Prize from IEEE Control Systems Society, the Presidential Early Career Award for Scientists and Engineers (PECASE), the National Science Foundation (NSF) CAREER award, and the IEEE 2012 Outstanding Engineer Award of Region 6, among other awards. Her research is on mobile sensor networks. Current research thrusts include X-ray vision for robots, RF sensing, communication-aware robotics, occupancy estimation, see-through imaging, and human-robot collaboration. Her research has appeared in several reputable news venues such as BBC, Huffington Post, Daily Mail, Engadget, and NSF Science360, among others. She currently serves on the Board of Governors (BoG) for IEEE Control Systems Society (CSS) and is also a Senior Editor for the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS. She is a fellow of IEEE.