Deep Learning for Detecting Robotic Grasps

Ian Lenz,† Honglak Lee,* and Ashutosh Saxena†
† Department of Computer Science, Cornell University.
* Department of EECS, University of Michigan, Ann Arbor.
Email: ianlenz@cs.cornell.edu, honglak@eecs.umich.edu, asaxena@cs.cornell.edu

***Abstract*—We consider the problem of detecting robotic grasps in an RGB-D view of a scene containing objects. In this work, we apply a deep learning approach to solve this problem, which avoids time-consuming hand-design of features. This presents two main challenges. First, we need to evaluate a huge number of candidate grasps. In order to make detection fast, as well as robust, we present a two-step cascaded structure with two deep networks, where the top detections from the first are re-evaluated by the second. The first network has fewer features, is faster to run, and can effectively prune out unlikely candidate grasps. The second, with more features, is slower but has to run only on the top few detections. Second, we need to handle multimodal inputs well, for which we present a method to apply structured regularization on the weights based on multimodal group regularization. We demonstrate that our method outperforms the previous state-of-the-art methods in robotic grasp detection.**

I. Introduction

Robotic grasping is a challenging problem involving perception, planning, and control. Some recent works [33, 35, 13, 41] address the perception aspect of this problem by converting it into a detection problem where, given a noisy, partial view of the object from a camera, the goal is to infer the top locations where a robotic gripper could be placed (see Figure 1). Unlike generic vision problems based on static images, such robotic perception problems are often used in closed loop with controllers, so there are stringent requirements on performance and computational speed. In the past, hand-designing these features has been the most popular method for several robotic tasks, such as [23, 18]. However, this is cumbersome and time-consuming, especially when we must incorporate new input modalities such as RGB-D cameras.

Recent methods based on deep learning [1] have demonstrated state of the art performance in a wide variety of tasks, including visual recognition [19, 37], audio recognition [22, 25], and natural language processing [6]. These techniques are especially powerful because they are capable of learning useful features directly from unlabeled and labeled data, avoiding the need for hand-engineering.

However, most work in deep learning has been applied in the context of *recognition*. Grasping is inherently a *detection* problem (see Figure 1), and previous applications of deep learning to detection have typically focused on specific applications such as face detection [27]. Our goal is not only to infer a viable grasp, but to infer the optimal grasp for a given object that maximizes the chance of successfully grasping it. Thus, the first major contribution of our work is to apply deep learning to the problem of robotic grasping, in a fashion which would generalize to similar detection problems.

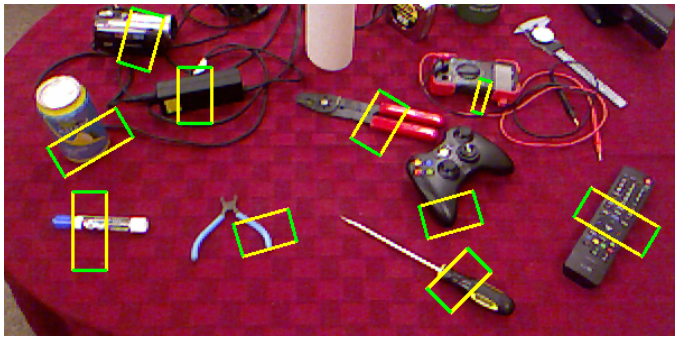


Fig. 1: **Detecting robotic grasps.** A cluttered lab scene labeled with rectangles corresponding to robotic grasps for objects in the scene. Green lines correspond to robotic gripper plates. We use a two-stage system based on deep learning to learn features and perform detection for robotic grasping.

The second major contribution of our work is to propose a new method for handling multimodal data in the context of feature learning. The use of RGB-D data, as opposed to simple 2D image data, has been shown to significantly improve grasping detection results [13, 7, 35]. In this work, we present a multimodal feature learning algorithm which adds a structured regularization penalty to the objective function to be optimized during learning. As opposed to previous works in deep learning, which either ignore modality information at the first layer, encouraging all features to use all modalities [36], or train separate first-layer features for each modality [26, 39], our approach allows for a middle-ground in which each feature is encouraged to use only a subset of the input modalities, but is not forced to use only particular ones.

We also propose a two-stage cascaded deep learning detection system, where we have fewer features in the first layer that provide faster but only approximately accurate detections, and more features in the second layer that provide more accurate detections. In our experiments, we found that the first deep network with fewer features was better at avoiding overfitting but less accurate. We feed the top rectangles from the first layer into the second layer, leading to robust early rejection of false positives. Unlike manually designed two-step features as in [13], our method uses deep learning, which allows us to learn detectors that not only give higher performance, but are also computationally efficient.

We test our approach on a challenging dataset, where we show that our algorithm improves both recognition and detection performance for grasping rectangle data. We also show that our two-stage approach is not only able to match the performance of a single-stage system using a larger feature

set, but in fact, improves results, while significantly reducing the computational time needed for detection.

In summary, the contributions of this paper are:

- We present a deep learning algorithm for detecting robotic grasps. To the best of our knowledge, this is the first work to do so.
- In order to handle multi-modal inputs, we present a new way to apply structured regularization to the weights to these inputs based on multimodal group sparsity.
- In order to improve computational performance, we present multi-step cascaded structure for detection, significantly reducing the computational cost of grasp detection.
- Our method outperforms the state-of-the-art of grasp detection, as well as previous deep learning algorithms.

The rest of the paper is organized as follows: We discuss related work in Section II. We present our two-step cascaded detection system in Section III. We then describe our feature learning algorithm and structured regularization method in Section IV. We present our experiments in Section V, and show and discuss results in Section VI. We conclude in Section VII.

II. Related Work

Deep Learning. A handful of previous works have applied deep learning to detection problems [27, 21, 5]. For example, Osadchy et al. [27] applied a deep energy-based model to the problem of face detection, and Coates et al. [5] used a deep learning approach to detect text in images. Both of these problems differ significantly from robotic grasp detection. In each, an image contains some set of true detections, and the goal is to find *all* of them, while in robotic grasp detection, an image might contain a large number of possible grasps, and the goal is to find the *best* one, requiring a different approach.

Jalali et al. [12] used a structured regularization function similar to that which we propose here. However, their work applies it only to *multitask learning* problems (multiple linear regression tasks), while we apply it to more complex non-linear deep networks in a very different context of *multimodal learning*. The Topographic ICA algorithm [11] is a feature-learning approach that applies a structured penalty term to feature activations, but not to the weights themselves.

Coates and Ng [4] investigate the problem of selecting receptive fields, i.e., subsets of the input features to be used together in a higher-level feature. In this paper, we apply similar concepts to multi-modal features via structured regularization.

Previous works on multimodal deep learning have focused on learning separate features for modalities with significantly different representations. Ngiam et al. [26] worked with audio and video data, while Srivastava and Salakhutdinov [39] worked with images and text. Our work proposes an algorithm which would apply both to these cases and to cases such as RGBD data, where the underlying representation of the modalities is similar. Previous work on RGBD recognition (e.g., [36]) typically ignores correlated modality information and simply concatenates features from each modality.

Robotic Grasping. Many works focus on determining feasible grasps given full knowledge of 2D or 3D object shape using physics-based techniques such as force- and form-closure [29]. Some recent works [7, 9] use full physical simulation given 3D models to determine feasible grasps. Gallegos et al. [8] performed optimization of grasps given both a 3D model of the object to be grasped and the desired contact points for the robotic gripper. Our approach requires only a single RGBD view and thus can be applied to cases where the full 3D model of an object is not known.

Other methods focus on specific cases of robotic grasping. For example, [24, 3, 28] assume that objects to be grasped belong to a particular set of shape primitives or compositions thereof. Our approach is able to learn features and detect feasible grasps regardless of object shape.

Learning based methods have enabled grasp detection to generalize to novel objects [33]. However, all previous image-based approaches to grasping novel objects have used hand-designed features. Some works rely exclusively on 2D image features such as edge and texture features [34]. However, most recent works combine 2D and 3D features, either using similar features for both [13, 14], or extracting geometric information from 3D data [20, 30, 35].

These works typically consider only one type of gripper, either two-fingered/parallel-plate [13, 38], three-finger [20, 30], or jamming [14]. Some works consider multiple types [34, 35], but use the same features for all. Here, we will consider parallel plate grippers, but in the future, our approach could be used to learn gripper-specific features for any of these types.

RGB-D Data. Due to the availability of inexpensive depth sensors, RGB-D data has been a significant research focus in recent years for various applications. For example, Jiang et al. [15] consider robotic placement of objects, Koppula et al. [17] consider human activity detection, and Koppula et al. [16] consider object detection in 3D scenes. Most works with RGB-D data use hand-engineered features such as [32]. The few works that perform feature learning for RGB-D data [36, 2] largely ignore the multimodal nature of the data, not distinguishing the color and depth channels. Here, we present a structured regularization approach which allows us to learn more robust features for RGB-D and other multimodal data.

III. Deep Learning for Grasp Detection: System and Model

In our system for robotic grasping, the robot first obtains an RGB-D image of the scene containing objects to be grasped. A small deep network is used to score potential grasps in this image, and a small candidate set of the top-ranked grasps is provided to a larger deep network, which yields a single best-ranked grasp. The robot then uses the parameters of this detected grasp to plan a path and grasp the object. We will represent potential grasps using oriented rectangles in the image plane, with one pair of parallel edges corresponding to the robotic gripper [13].

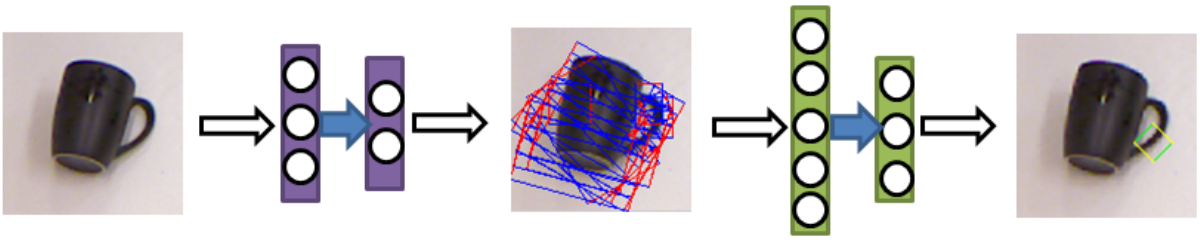


Fig. 2: **Illustration of our two-stage detection process.** Given an image of an object to grasp, a small deep network is used to exhaustively search potential rectangles, producing a small set of top-ranked rectangles. A larger deep network is then used to find the top-ranked rectangle from these candidates, producing a single optimal grasp for the given object.

Using a standard feature learning approach such as sparse auto-encoder [10], a deep network can be trained for the problem of grasping rectangle recognition (i.e., does a given rectangle in image space correspond to a valid robotic grasp?). However, in a real-world robotic setting, our system needs to perform *detection* (i.e., given an image containing an object, how should the robot grasp it?). This task is significantly more challenging than simple recognition.

Two-stage Cascaded Detection. In order to perform detection, one naive approach could be to consider each possible oriented rectangle in the image (perhaps discretized to some level), and evaluate each rectangle with a deep network trained for recognition. However, such near-exhaustive search of possible rectangles (based on positions, sizes, and orientations) can be quite expensive in practice for real-time robotic grasping.

Motivated by multi-step cascaded approaches in previous work [13, 40], we instead take a two-stage approach to detection: First, we use a reduced feature set to determine a set of top candidates. Then, we use a larger, more robust feature set to rank these candidates.

However, these approaches require the design of two separate sets of features. In particular, it can be difficult to manually design a small set of first-stage features which is both quick to compute and robust enough to produce a good set of candidate detections for the second stage. Using deep learning allows us to circumvent the costly manual design of features by simply training networks of two different sizes, using the smaller for the exhaustive first pass, and the larger to re-rank the candidate detection results.

Model. To detect robotic grasps from the rectangle representation, we model the probability of a rectangle $G^{(t)}$, with features $x^{(t)} \in \mathbb{R}^N$ being graspable, using a random variable $\hat{y}^{(t)} \in \{0, 1\}$ which indicates whether or not we predict $G^{(t)}$ to be graspable. We use a deep network with two layers of sigmoidal hidden units $h^{[1]}$ and $h^{[2]}$, with K_1 and K_2 units per layer, respectively. A logistic classifier over the outputs second-layer hidden units then predicts $P(\hat{y}^{(t)}|x^{(t)}; \Theta)$. Each layer ℓ will have a set of weights $W^{[\ell]}$ mapping from its inputs to its hidden units, so the parameters of our model are $\Theta = (W^{[1]}, W^{[2]}, W^{[3]})$. Each hidden unit forms output by a sigmoid $\sigma(a) = 1/(1 + \exp(-a))$ over its weighted input:

$$h_j^{[1](t)} = \sigma\left(\sum_{i=1}^{N} x_i^{(t)} W_{i,j}^{[1]}\right)$$

$$h_j^{[2](t)} = \sigma\left(\sum_{i=1}^{K_1} h_i^{[1](t)} W_{i,j}^{[2]}\right)$$

$$P(\hat{y}^{(t)} = 1|x^{(t)}; \Theta) = \sigma\left(\sum_{i=1}^{K_2} h_i^{[2](t)} W_i^{[3]}\right)$$

A. Inference and Learning

During **inference**, our goal is to find the single grasping rectangle with the maximum probability of being graspable for some new object. With G representing a particular grasping rectangle position, orientation, and size, we find this best rectangle as:

$$G^* = \arg\max_G \; P(\hat{y}^{(t)} = 1|\phi(G); \Theta)$$

Here, the function ϕ extracts the appropriate input representation for rectangle G.

During **learning**, our goal is to learn the parameters Θ that optimize the recognition accuracy of our system. Here, input data is given as a set of pairs of features $x^{(t)} \in \mathbb{R}^N$ and ground-truth labels $y^{(t)} \in \{0, 1\}$ for $t = 1, \ldots, M$. As in most deep learning works, we use a two-phase learning approach.

In the first phase, we will use *unsupervised feature learning* to initialize the hidden-layer weights $W^{[1]}$ and $W^{[2]}$. Pre-training weights this way is critical to avoid overfitting. We will use a variant of the sparse auto-encoder (SAE) algorithm [10]. We define $g(h)$ as a sparsity penalty function over hidden unit activations, with λ controlling its weight. If $f(W)$ is a regularization function, weighted by β, and $\hat{x}^{(t)}$ is a reconstruction of $x^{(t)}$, SAE solves the following to initialize hidden-layer weights:

$$W^* = \arg\min_W \sum_{t=1}^{M} ||\hat{x}^{(t)} - x^{(t)}||_2^2 + \lambda \sum_{j=1}^{K} g(h_j^{(t)}) + \beta f(W)$$

$$\hat{x}_i^{(t)} = \sum_{j=1}^{K} h_j^{(t)} W_{i,j} \tag{1}$$

We first use this algorithm to intialize $W^{[1]}$ to reconstruct x. We then fix $W^{[1]}$ and learn $W^{[2]}$ to reconstruct $h^{[1]}$.

During the *supervised* phase of the learning algorithm, we then jointly learn classifier weights $W^{[3]}$ and fine-tune hidden layer weights $W^{[1]}$ and $W^{[2]}$ for recognition. We maximize the log-likelihood of the data along with regularization penalties

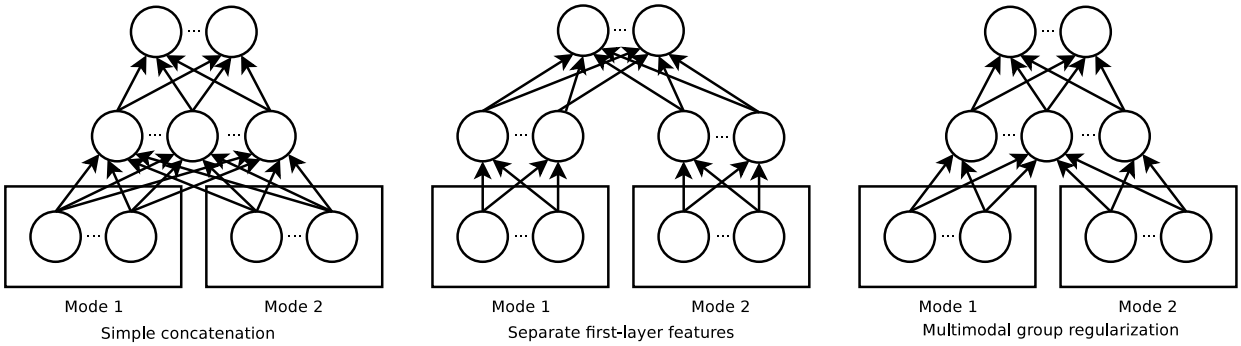


Fig. 3: **Three possible models for multimodal deep learning.** Left: fully dense model - all visible features are concatenated and modality information is ignored. Middle: modality-specific sparse model - separate first layer features are trained for each modality. Right: group-sparse model - a structured regularization term encourages features to use only a subset of the input modes.

on hidden layer weights:

$$\Theta^* = \arg\max_{\Theta} \sum_{t=1}^{M} \log P(\hat{y}^{(t)} = y^{(t)} | x^{(t)}; \Theta) - \beta_1 f(W^{[1]}) - \beta_2 f(W^{[2]}) \quad (2)$$

Two-stage Detection Model. During **inference** for two-stage detection, we will first use a smaller network to produce a set of the top T rectangles with the highest probability of being graspable according to network parameters Θ_1. We will then use a larger network with a separate set of parameters Θ_2 to re-rank these T rectangles and obtain a single best one. The only change to **learning** for the two-stage model is that these two sets of parameters are learned separately, using the same approach.

IV. Structured Regularization for Feature Learning

In the multimodal setting, we assume that the input data x is known to come from R distinct modalities, for example audio and video data, or depth and RGB data. We define the modality matrix S as an RxN binary matrix, where each element $S_{r,i}$ indicates membership of visible unit x_i in a particular modality r, such as depth or image intensity.

A naive way of applying feature learning to this data is to simply take x (as a concatenated vector) as input to the model described above, ignoring information about specific modalities, as seen on the lefthand side of Figure 3. This approach may either 1) prematurely learn features which include all modalities, which can lead to overfitting, or 2) fail to learn associations between modalities with very different underlying statistics.

Instead of concatenating multimodal input as a vector, Ngiam et al. [26] proposed training a first layer representation for each modality separately, as shown in Figure 3-middle. This approach makes the assumption that the ideal low-level features for each modality are purely unimodal, while higher-layer features are purely multimodal. This approach may work better for some problems where the modalities have very different basic representations, such as the video and audio data (as used in [26]), so that separate first layer features may give better performance. However, for modalities such as RGBD data, where the input modes represent different channels of an image, learning low-level correlations can lead to more robust features – our experiments in Section V show that simply concatenating the input modalities significantly outperforms training separate first-layer features for robotic grasp detection from RGBD data.

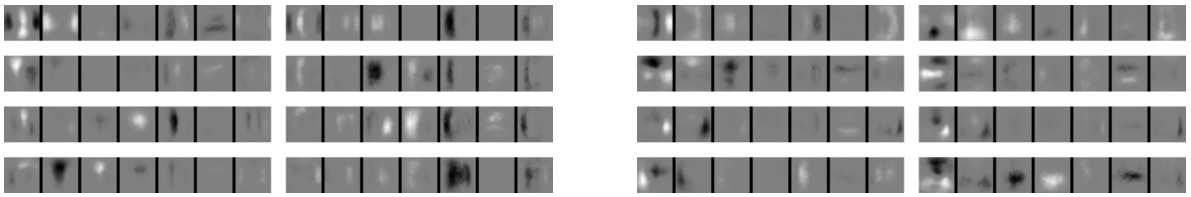
For many problems, it may be difficult to tell which of these approaches will perform better, and time-consuming to tune and comparatively evaluate multiple algorithms. In addition, the ideal feature set for some problems may contain features which use some, but not all, of the input modalities, a case which neither of these approaches are designed to handle.

To solve these problems, we propose a new algorithm for feature learning for multimodal data. Our approach incorporates a structured penalty term into the optimization problem to be solved during learning. This technique allows the model to learn correlated features between multiple input modalities, but regularizes the number of modalities used per feature (hidden unit), discouraging the model from learning weak correlations between modalities. With this regularization term, the algorithm can specify how mode-sparse or mode-dense the features should be, representing a continuum between the two extremes outlined above.

Regularization in Deep Learning. In a typical deep learning model, L_1 regularization (i.e., $f(W) = ||W||_1$) or L_2 regularization (i.e., $f(W) = ||W||_2^2$) are commonly used in training (e.g., as specified in Equations (1) and (2)). These are often called a "weight cost" (or "weight decay"), and are left implicit in many works.

Applying regularization is well known to improve the generalization performance of feature learning algorithms. One might expect that a simple L_1 penalty would eliminate weak correlations in multimodal features, leading to features which use only a subset of the modes each. However, we found that in practice, a value of β large enough to cause this also degraded the quality of features for the remaining modes and lead to decreased task performance.

Multimodal Regularization. For structured multimodal regularization, each modality will be used as a regularization group separately for each hidden unit, applied in a manner similar



(a) Features corresponding to positive grasps.

(b) Features corresponding to negative grasps.

Fig. 4: **Features learned from grasping data.** Each feature contains seven channels - from left to right, depth, Y, U, and V image channels, and X, Y, and Z surface normal components. Vertical edges correspond to gripper plates. Left: eight features with the strong positive correlations to rectangle graspability. Right: similar, but negative correlations. Group regularization eliminates many modalities from many of these features, making them more robust.

to the group regularization in [12]:

$$f(W) = \sum_{j=1}^{K} \sum_{r=1}^{R} \left(\sum_{i=1}^{N} S_{r,i} |W_{i,j}^{p}| \right)^{1/p} \tag{3}$$

Using a high value of p allows us to penalize higher-valued weights from each mode to each feature more strongly than lower-valued ones. At the limit ($p \to \infty$), this group regularization becomes equivalent to the infinity (or max) norm:

$$f(W) = \sum_{j=1}^{K} \sum_{r=1}^{R} \max_{i} S_{r,i} |W_{i,j}| \tag{4}$$

which penalizes only the maximum weight from each mode to each feature. In practice, the infinity norm is not differentiable and therefore is difficult to apply gradient-based optimization methods; in this paper, we use the log-sum-exponential as a differentiable approximation to the max norm.

In experiments, this regularization function produces first-layer weights concentrated in fewer modes per feature. However, we found that at values of β sufficient to induce the desired mode-wise sparsity patterns, penalizing the maximum also had the undesirable side-effect of causing many of the weights for other modes to saturate at their mode's maximum, suggesting that the features were overly constrained. In some cases, constraining the weights in this manner also caused the algorithm to learn duplicate (or redundant) features, in effect scaling up the feature's contribution to reconstruction to compensate for its constrained maximum. This is obviously an undesirable effect, as it reduces the effective size (or diversity) of the learned feature set.

This suggests that the max-norm may be overly constraining. A more desirable sparsity function would penalize nonzero weight maxima for each mode for each feature without additional penalty for larger values of these maxima. We can achieve this effect by applying the L_0 norm, which takes a value of 0 for an input of 0, and 1 otherwise, on top of the max-norm from above:

$$f(W) = \sum_{j=1}^{K} \sum_{r=1}^{R} \mathbb{I}\{(\max_{i=1} S_{r,i} |W_{i,j}|) > 0\} \tag{5}$$

where $\mathbb{I}$ is the indicator function, which takes a value of 1 if its argument is true, 0 otherwise. Again, for a gradient-based method, we used an approximation to the L_0 norm, such as $\log(1+x^2)$. This regularization function now encodes a direct penalty on the number of modes used for each weight, without further constraining the weights of modes with nonzero maxima.

Fig. 5: **Example objects from the Cornell grasping dataset.** [13]. This dataset contains objects from a large variety of categories.

Figure 4 shows features learned from the unsupervised stage of our group-regularized deep learning algorithm. We discuss these features, and their implications for robotic grasping, in Section VI.

V. Experiments

Dataset. We used the extended version of the Cornell grasping dataset [13] for our experiments (http://pr.cs.cornell.edu/deepgrasping). This dataset contains 1035 images of 280 graspable objects, each annotated with several ground-truth positive and negative grasping rectangles. While the vast majority of possible rectangles for most objects will be non-graspable, the dataset contains roughly equal numbers of graspable and non-graspable rectangles. We will show that this is useful for an unsupervised learning algorithm, as it allows learning a good representation for graspable rectangles even from unlabeled data.

We performed five-fold cross-validation, and present results for splits on a per image (i.e., the training set and the validation set do not share the same image) and per object (i.e., the training set and the validation set do not share any images from the same object) basis.

We take seven channels as input: YUV channels in the color space, depths, and the XYZ components of computed surface normals. With an image patch size of 24x24 pixels, we have 4032 (=24*24*7) input features. We trained a deep network with 200 hidden units each at the first and second layers using our learning algorithm as described in Sections III and IV,

Preserving Aspect Ratio. It is important to preserve aspect ratio when feeding features into the network. However,

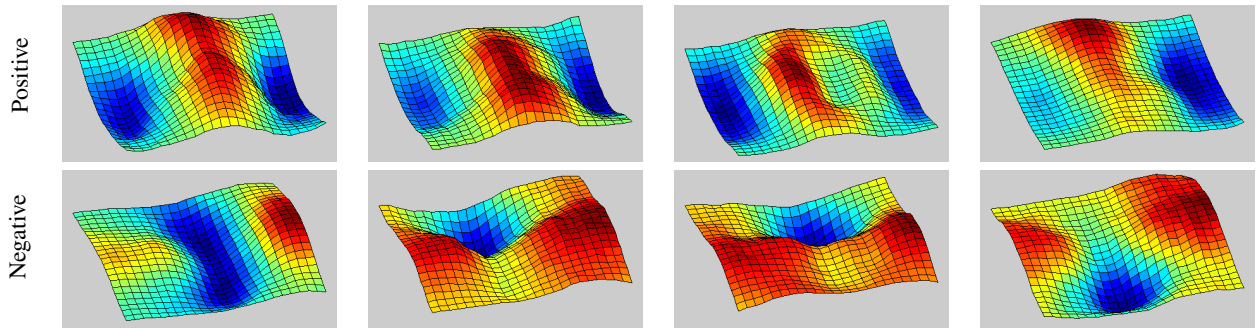


Fig. 6: **Learned 3D depth features.** 3D meshes for depth channels of the four features with strongest positive (top) and negative(bottom) correlations to rectangle graspability. Here X and Y coordinates corresponds to positions in the deep network's receptive field, and Z coordinates corresponds to weight values to the depth channel for each location. Feature shapes clearly correspond to graspable and non-graspable structures, respectively.

padding with zeros can bias the network towards square rectangles which fill its receptive field and thus give more nonzero inputs. To address this problem, we define a multiplicative scaling factor for the inputs from each modality, based on the fraction of each mode which is masked out: $\Psi_l^{(t)} = \sum_{i=1}^{N} S_{l,i} / \left(\sum_{i=1}^{N} S_{l,i} \mu_i^{(t)} \right)$, where $\mu_i^{(t)}$ is 1 if $x_i^{(t)}$ is masked in, 0 otherwise.[1] In practice, we found it necessary to limit the scaling factor to a maximum of some value c, as $\Psi'^{(t)}_l = \min(\Psi_l^{(t)}, c)$.

Baselines. We compare our recognition results in the Cornell grasping dataset with the features from [13], as well as the combination of these features and Fast Point Feature Histogram (FPFH) features [31]. We used a linear SVM for classification, which gave the best results among all other kernels.

We also compare our algorithm to other deep learning approaches. We compare to a network trained only with standard L1 regularization, and a network trained in a manner similar to [26], where three separate sets of first layer features are learned for the depth channel, the combination of the Y, U, and V channels, and the combination of the X, Y, and Z surface normal components.

Metrics for Detection. For detection, we compare the top-ranked rectangle for each method with the set of ground-truth rectangles for each image. We present results using two metrics, the "point" and "rectangle" metric.

For the point metric, similar to [34], we compute the center point of the predicted rectangle, and consider the grasp a success if it is within some distance from at least one ground-truth rectangle center. We note that this metric ignores grasp orientation, and therefore might overestimate the performance of an algorithm for robotic applications.

For the rectangle metric, similar to [13], let G be the top-ranked grasping rectangle predicted by the algorithm, and G^* be a ground-truth rectangle. Any rectangles with an orientation error of more than 30^o from G are rejected. From the remaining set, we use the common bounding box evaluation metric of intersection divided by union - i.e. $Area(G \cap G^*)/Area(G \cup G^*)$. Since a ground-truth rectangle can define a large space of graspable rectangles (e.g., covering the entire length of a pen), we consider a prediction to be correct if it scores at least 25% by this metric.

[1]Implementation detail: Since we use the squared reconstruction error, we found that simply scaling the input caused the learning algorithm to put too much significance to cases where more data is masked out. As a heuristic to address this issue, when pretraining with SAE, we scaled the input to the network and the reconstruction penalty for each input coordinate, but not the target value for reconstruction.

VI. Results and Discussion

A. Deep Learning for Robotic Grasp Detection

Figure 4 shows the features learned by the unsupervised phase of our algorithm which have a high correlation to positive and negative grasping cases. Many of these features show non-zero weights to the depth channel, indicating that it learns the correlation of depths to graspability. Figure 6 shows 3D meshes for the depth channels of the four features with the strongest positive and negative correlations to valid grasps. Even *without any supervised information*, our algorithm was able to learn several features which correlate strongly to graspable cases and non-graspable cases. The first two positive-correlated features represent handles, or other cases with a raised region in the center, while the second two represent circular rims or handles. The negatively-correlated features represent obviously non-graspable cases, such as ridges perpendicular to the gripper plane and "valleys" between the gripper plates. From these features, we can see that even during unsupervised feature learning, our approach is able to learn a task-specific representation.

From Table I, we see that the recognition performance is significantly improved with deep learning methods, improving 9% over the features from [13] and 4.1% over those features combined with FPFH features. Both L_1 and group regularization performed similarly for recognition, but training separate first layer features decreased performance slightly.

Table II shows that, once mask-based scaling has been applied, all deep learning approaches except for training separate first-layer features outperform the hand-engineered features from [13] by up to 13% for the point metric and 17% for the rectangle metric, while also avoiding the need to design task-specific features.

Adaptability. One important advantage of our detection system is that we can flexibly specify the constraints of the gripper in our detection system. Different robots have different

TABLE I: **Recognition results for Cornell grasping dataset.**

Algorithm	Accuracy (%)
Jiang et al. [13]	84.7
Jiang et al. [13] + FPFH	89.6
Sparse AE, separate layer-1 feat.	92.8
Sparse AE	**93.7**
Sparse AE, group reg.	**93.7**

TABLE II: **Detection results for point and rectangle metrics**, for our variants of sparse auto-encoders.

Algorithm	Image-wise split		Object-wise split	
	Point	Rect	Point	Rect
Jiang et al. [13]	75.3	60.5	74.9	58.3
SAE, no mask-based scaling	62.1	39.9	56.2	35.4
SAE, separate layer-1 feat.	70.3	43.3	70.7	40.0
SAE, L_1 reg.	87.2	72.9	**88.7**	71.4
SAE, struct. reg., 1^{st} pass only	86.4	70.6	85.2	64.9
SAE, struct. reg., 2^{nd} pass only	87.5	73.8	87.6	73.2
SAE, struct. reg. two-stage	**88.4**	**73.9**	88.1	**75.6**

gripper—PR2 has a wide gripper, and Adept Viper arm has a smaller one. We can constrain the detectors to handle this. Figure 7 shows detection scores for systems constrained based on the PR2 and Adept grippers. For grippers with different properties, such as multi-fingered or jamming grippers, our algorithm would be able to learn new features for detection given only data labeled for the desired gripper.

B. Multimodal Group Regularization.

Our group regularization term improves detection accuracy over simple L_1 regularization. The improvement is more significant for the object-wise split than for the image-wise split because the group regularization helps the network to avoid overfitting, which will tend to occur more when the learning algorithm is evaluated on unseen objects.

Figure 8 shows typical cases where a network trained using our group regularization finds a valid grasp, but a network trained with L_1 regularization does not. In these cases, the grasp chosen by the L_1-regularized network appears valid for some modalities – the depth channel for the sunglasses and nail polish bottle, and the RGB channels for the scissors. However, when all modalities are considered, the grasp is clearly invalid. The group-regularized network does a better job of combining information from all modalities and is more robust to noise and missing data in the depth channel, as seen in these cases.

C. Two-stage Detection System.

We tested our two-stage system by training a network with 50 hidden units at the first and second layers. Learning and detection were performed in the same manner as with the full-size network, except that the top 100 rectangles for each image were recorded, then re-ranked using the full-size network to yield a single best-scoring rectangle. The number of rectangles the full-size network needed to evaluate was reduced by roughly a factor of 1000.

Using our two-stage approach increased detection performance up to 2% as compared to a single pass with the large-size network, even though using the small network alone significantly underperforms the larger network. In most cases, the top 100 rectangles from the first pass contained the top-ranked rectangle from an exhaustive search using the second-stage network, and thus results were unaffected.

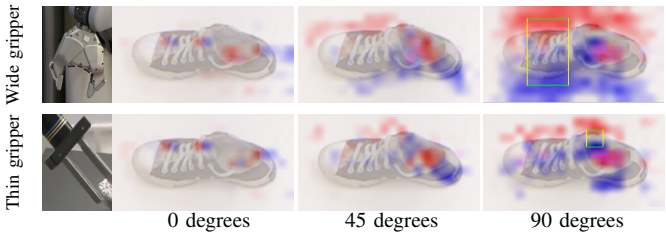


Fig. 7: **Visualization of grasping scores for different grippers.** Red indicates maximum score for a grasp with left gripper plane centered at each point, blue is similar for the right plate. Best-scoring rectangle shown in green/yellow.

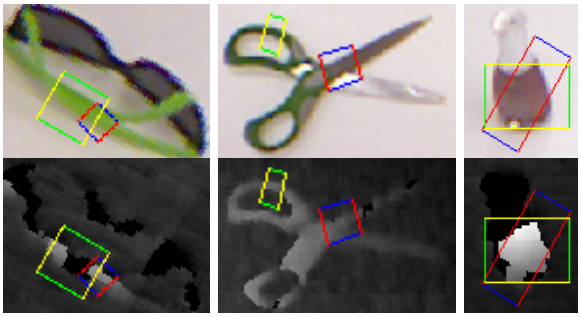


Fig. 8: **Improvements from group regularization.** Cases where our group regularization approach produces a viable grasp (shown in green and yellow), while a network trained only with simple L_1 regularization does not (shown in blue and red). Top: RGB image, bottom: depth channel. Green and blue edges correspond to gripper.

Figure 9 shows some cases where the first-stage network pruned away rectangles corresponding to weak grasps which might otherwise be chosen by the second-stage network. In these cases, the grasp chosen by the single-stage system might be feasible for a robotic gripper, but the rectangle chosen by the two-stage system represents a grasp which would clearly be successful.

The two-stage system also significantly increases the computational efficiency of our detection system. Average inference time for a MATLAB implementation of the deep network was reduced from 24.6s/image for an exhaustive search using the larger network to 13.5s/image using the two-stage system.

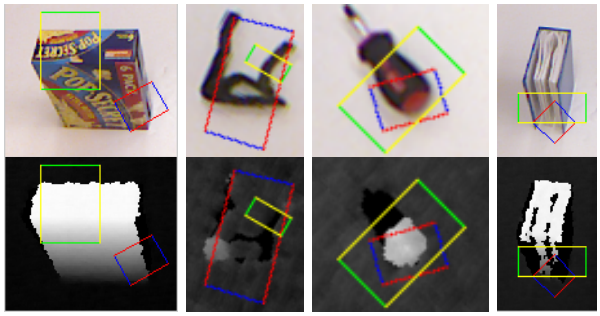


Fig. 9: **Improvements from two-stage system.** Example cases where the two-stage system produces a viable grasp (shown in green and yellow), while the single-stage system does not (shown in blue and red). Top: RGB image, bottom: depth channel. Green and blue edges correspond to gripper.

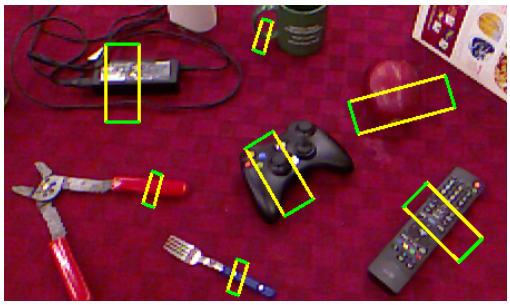


Fig. 10: **Multiple grasp detection.** – A cluttered scene with many graspable objects for which our system identifies valid grasps.

Finally, Figure 10 shows the result of our detection in a multi-object scenario. Given segmentation from the background, our algorithm is able to detect grasps for a wide variety of objects, even those such as the XBox controller which were not present in the dataset.

VII. Conclusions

We presented a system for detecting robot grasps from RGBD data using a deep learning approach. Our method has several advantages over current state-of-the-art methods. First, using deep learning allows us to avoid hand-engineering features, learning them instead. Second, our results show that deep learning methods significantly outperform even well-designed hand-engineered features from previous work.

We also presented a novel feature learning algorithm for multimodal data based on group regularization. In extensive experiments, we demonstrated that this algorithm produces better features for robotic grasp detection than existing deep learning approaches to multimodal data. Our experiments and results show that our two-stage deep learning system with group regularization is capable of robustly detecting grasps for a wide range of objects, even those previously unseen by the system.

Many robotics problems require the use of perceptual information, but can be difficult and time-consuming to engineer good features for. In future work, our approach could be extended to a wide range of such problems.

Acknowledgements

We would like to thank Yun Jiang and Marcus Lim for useful discussions and help with baseline experiments. This research was funded in part by ARO award W911NF-12-1-0267, Microsoft Faculty Fellowship and NSF CAREER Award (Saxena), and Google Faculty Research Award (Lee).

References

[1] Y. Bengio. Learning deep architectures for AI. *FTML*, 2(1):1–127, 2009.
[2] L. Bo, X. Ren, and D. Fox. Unsupervised Feature Learning for RGB-D Based Object Recognition. In *ISER*, 2012.
[3] D. Bowers and R. Lumia. Manipulation of unmodeled objects using intelligent grasping schemes. *IEEE Trans Fuzzy Sys*, 11(3), 2003.
[4] A. Coates and A. Y. Ng. Selecting receptive fields in deep networks. In *NIPS*, 2011.
[5] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu, and A. Y. Ng. Text detection and character recognition in scene images with unsupervised feature learning. In *ICDAR*, 2011.
[6] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12:2493–2537, 2011.
[7] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa. Physics-based grasp planning through clutter. In *RSS*, 2012.
[8] C. R. Gallegos, J. Porta, and L. Ros. Global optimization of robotic grasps. In *RSS*, 2011.
[9] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen. The Columbia grasp database. In *ICRA*, 2009.
[10] I. Goodfellow, Q. Le, A. Saxe, H. Lee, and A. Y. Ng. Measuring invariances in deep networks. In *NIPS*, 2009.
[11] A. Hyvärinen, P. O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural computation*, 13(7):1527–1558, 2001.
[12] A. Jalali, P. Ravikumar, S. Sanghavi, and C. Ruan. A dirty model for multi-task learning. In *NIPS*, 2010.
[13] Y. Jiang, S. Moseson, and A. Saxena. Efficient grasping from RGBD images: Learning using a new rectangle representation. In *ICRA*, 2011.
[14] Y. Jiang, J. R. Amend, H. Lipson, and A. Saxena. Learning hardware agnostic grasps for a universal jamming gripper. In *ICRA*, 2012.
[15] Y. Jiang, M. Lim, C. Zheng, and A. Saxena. Learning to place new objects in a scene. *IJRR*, 31(9), 2012.
[16] H. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*, 2011.
[17] H. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from rgb-d videos. *IJRR*, 2013.
[18] D. Kragic and H. I. Christensen. Robust visual servoing. *IJRR*, 2003.
[19] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In *ICML*, 2012.
[20] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng. Learning to grasp objects with multiple contact points. In *ICRA*, 2010.
[21] Y. LeCun, F. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR*, 2004.
[22] H. Lee, Y. Largman, P. Pham, and A. Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *NIPS*, 2009.
[23] J. Maitin-shepard, M. Cusumano-towner, J. Lei, and P. Abbeel. Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *ICRA*, 2010.
[24] A. T. Miller, S. Knoop, P. K. Allen, and H. I. Christensen. Automatic grasp planning using shape primitives. In *ICRA*, 2003.
[25] A.-R. Mohamed, G. Dahl, and G. E. Hinton. Acoustic modeling using deep belief networks. *IEEE Trans Audio, Speech, and Language Processing*, 20(1):14–22, 2012.
[26] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, 2011.
[27] M. Osadchy, Y. LeCun, and M. Miller. Synergistic face detection and pose estimation with energy-based models. *JMLR*, 8:1197–1215, 2007.
[28] J. H. Piater. Learning visual features to predict hand orientations. In *ICML*, 2002.
[29] J. Ponce, D. Stam, and B. Faverjon. On computing two-finger force-closure grasps of curved 2D objects. *IJRR*, 12(3):263, 1993.
[30] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng. Grasping novel objects with depth segmentation. In *IROS*, 2010.
[31] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *ICRA*, 2009.
[32] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3D recognition and pose using the viewpoint feature histogram. In *IROS*, 2010.
[33] A. Saxena, J. Driemeyer, J. Kearns, and A. Ng. Robotic grasping of novel objects. In *NIPS*, 2006.
[34] A. Saxena, J. Driemeyer, and A. Y. Ng. Robotic grasping of novel objects using vision. *IJRR*, 27(2):157–173, 2008.
[35] A. Saxena, L. L. S. Wong, and A. Y. Ng. Learning grasp strategies with partial shape information. In *AAAI*, 2008.
[36] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng. Convolutional-recursive deep learning for 3D object classification. In *NIPS*, 2012.
[37] K. Sohn, D. Y. Jung, H. Lee, and A. Hero III. Efficient learning of sparse, distributed, convolutional feature representations for object recognition. In *ICCV*, 2011.
[38] H. O. Song, M. Fritz, C. Gu, and T. Darrell. Visual grasp affordances from appearance-based cues. In *Robot Perception workshop ICCV*, 2011.
[39] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep Boltzmann machines. In *NIPS*, 2012.
[40] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.
[41] L. Zhang, M. Ciocarlie, and K. Hsiao. Grasp evaluation with graspable feature matching. In *RSS Workshop on Mobile Manipulation*, 2011.