Optimizing Correlated Graspability Score and Grasp Regression for Better Grasp Prediction

Amaury Depierre, Emmanuel Dellandréa, Liming Chen

Optimizing Correlated Graspability Score and Grasp Regression for Better Grasp Prediction

Amaury Depierre[1,2], Emmanuel Dellandréa[2] and Liming Chen[2]

***Abstract*— Grasping objects is one of the most important abilities to master for a robot in order to interact with its environment. Current state-of-the-art methods rely on deep neural networks trained to predict a graspability score jointly but separately from regression of an offset of grasp reference parameters, although the predicted offset could decrease the graspability score. In this paper, we extend a state-of-the-art neural network with a scorer which evaluates the graspability of a given position and introduce a novel loss function which correlates regression of grasp parameters with graspability score. We show that this novel architecture improves the performance from 81.95% for a state-of-the-art grasp detection network to 85.74% on Jacquard dataset. Because real-life applications generally feature scenes of multiple objects laid on a variable decor, we also introduce Jacquard+, a test-only extension of Jacquard dataset. Its role is to complete the traditional real robot evaluation by benchmarking the adaptability of a learned grasp prediction model on a different data distribution than the training one while remaining in totally reproducible conditions. Using this novel benchmark and evaluated through the Simulated Grasp Trial criterion, our proposed model outperforms a state-of-the-art one by 7 points.**

I. INTRODUCTION

Grasping objects is a crucial operation for many robot aided applications: industry and logistics, household, human interaction, *etc*. A reliable robotic grasp system could lead to a huge improvement in productivity as well as new applications. Yet the performances achieved by current state-of-the-art systems are far from what a human can do. Humans can pick objects -known or unknown- of any shape in dark or bright lighting conditions with a success rate close to 100%. To perform the same final action, a robotic system has to (1) analyze a sensor input to find a good grasp position, (2) plan a trajectory to reach this position, and (3) activate an end-effector to actually grasp the object. In this paper we focus on the first part of the process and more especially on detecting grasp positions for a parallel plate gripper in RGB images.

Early research works on grasp detection were based on analytic methods and 3D models [1] [2]. However, explicit object models are not always available in many real-world applications. The huge success of the paradigm of deep learning [3], especially in computer vision [4] [5], inspires researchers to use convolutional neural networks (CNN) for the problem of grasp prediction. In this case, the input of the network is a sensor image, either RGB, RGB-D, RGD (RGB image where the blue channel has been replaced by depth information) or depth only. Despite being more noisy than 3D models, these data are more readily available and thereby allow effective training of deep neural networks (DNN). DNN models achieved at the same time good performances [6] and real-time processing [7].

Fig. 1. Examples of images from Jacquard dataset (top row) and more challenging ones from the proposed Jacquard+ extension (bottom row)

The current literature feature two different kinds of approaches for grasp prediction using DNN: generation methods and evaluation methods. With generation methods, a DNN is trained to predict one (or multiple ordered) grasps from the sensor input. Predictions can be the direct regression of grasps parameters [7] or deformations values *w.r.t.* a reference grasp [8] [9] [10], also called anchor box. In evaluation methods, grasp proposals are presented to the network, whose role is to order them in terms of grasp quality [11] [12]. The proposed grasp detection method follows the generation approach which generates the grasp proposals and simultaneously evaluates their graspability. However, state-of-the-art generation oriented grasp detection methods, *e.g.*, [9] , predict from an input scene image a graspability score and regress simultaneously the offset of reference grasp parameters and don't explicitly correlate them in training. This could be problematic because a reference grasp position once moved with the regressed offset values could significantly decrease its graspability. In contrast, the proposed method proposes to explicitly correlate the regression of grasp parameters with its graspability score through a novel loss function.

To evaluate a grasp prediction model, it is generally tested through cross-validation on a training dataset and then transferred on a real robotic setup. As each physical setup is

[1]Siléane, Saint-Etienne, France a.depierre@sileane.com
[2]University of Lyon, Ecole Centrale de Lyon, LIRIS, CNRS UMR 5205, France {emmanuel.dellandrea, liming.chen}@ec-lyon.fr

unique (by its physical disposition, lighting conditions, tested objects *etc.*), it is not possible to fairly compare the results of multiple tests with each other.

In this paper, our contributions are threefold:

1) We propose a novel DNN architecture which uses its grasp quality evaluation to improve the grasp regression through a newly introduced loss;
2) We release Jacquard+, an extension of the Jacquard Dataset [13], which enables evaluation of grasp detection models on simulated scenes with multiple objects laid on a variable decor. It is created with physics simulation and allow test in totally reproducible conditions. Examples images can be seen on Fig. 1;
3) We perform extensive experiments and show that the proposed grasp detection methods significantly outperforms the state of the art both on Jacquard and its extension Jacquard+;

The rest of this paper is organized as follows. Section II overviews the related work. Section III explains the representation we used for robotic grasps. Section IV presents in details our network architecture and Section V the Jacquard+ dataset. Section VI discusses the experiments we made to evaluate our architecture and Section VII concludes this paper.

II. RELATED WORK

Past works [1] [2] in grasp detection used 3D models and analytical methods to analyse scenes and find grasp candidates. In real case applications with unknown objects or environment, a perfect knowledge of these models is difficult to acquire. For that reason, research was later oriented towards image analysis. More specifically, Lenz *et al.* [6] used a sliding window approach and a deep neural network to classify image patches and extract potential grasp locations. They reached an accuracy of 73.9% on the Cornell Grasping Dataset [1]. In [7], Redmon *et al.* improved both the accuracy and the processing time by using direct regression. Their network predicts 5 parameters describing the grasp as well as a positive outcome probability for each spatial area in a $N \times N$ grid covering the image. This method raised the accuracy to 88.0%.

Using deeper neural networks like ResNet [14] only marginally increased the performances in grasp detection to 89.21% in [15], motivating researchers to find an alternative to direct regression for grasp prediction. Inspired by work done in object detection [16] [17] [18], Guo *et al.* [8] and Chu *et al.* [10] introduced the notion of reference anchor box. With anchor boxes, the neural network is not trained to directly regress a grasp but instead to predict a deformation of a reference box. This simplifies the regression problem by introducing prior knowledge on the size of the expected grasps. In [8], the reference box is axis aligned and the network also produces a quality score and an orientation as classification between discrete angle values. In [10], the quality score is included in the orientation classification, letting the network predicting both grasp regression values and discrete orientation classification score. This network achieved 91.1% accuracy on Cornell Grasp Dataset.

[1] http://pr.cs.cornell.edu/grasping/rect_data/data.php

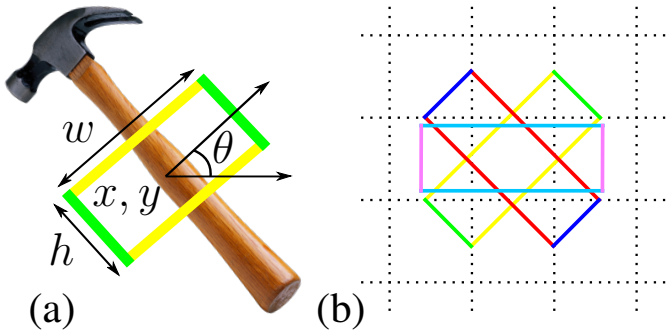

Fig. 2. (a) Example of a 5D representation of a grasp (b) Three examples of oriented anchor boxes with the same dimensions and angles of -45°, 0° and 45° centered on the same pixel of the features map

Zhou *et al.* proposed in [9] to remove the orientation classification by introducing oriented anchor boxes. Instead of having multiple scales or aspect ratios for the reference grasps, the authors used only one anchor box with multiple orientations. Then, the angle of the grasp is predicted in regard to this reference orientation, just like the position and size values. The network predicts therefore five regression values and one grasp quality score for each oriented reference box at each location in the feature map and achieves a state-of-the-art accuracy of 97.74%. However, grasp quality prediction only depends on the image information and is not directly correlated to the regression prediction. Our method extends this approach by adding a direct dependency between the regression prediction and the score evaluation.

III. PROBLEM STATEMENT

In this paper, we consider the problem of detecting successful grasp positions from RGB images of various objects lying on a plane. As the network does not have any space information, we do not use a 3D representation for the grasp but a 2D instead. Each grasp is represented as:

$$g = \{x, y, h, w, \theta\} \tag{1}$$

where (x, y) are position of the center, (h, w) its dimensions and θ its orientation as shown on Fig. 2 (a). In [6] Lenz *et al.* showed that this representation works well in practice, even with real robotic system.

To simplify the regression problem, prior knowledge about the position, size and orientation of the grasp are introduced through oriented reference grasp [9]. These reference grasps, also called anchor boxes, are defined as $g_a = (g_{ax}, g_{ay}, g_{aw}, g_{ah}, g_{a\theta})$. The grasp is then defined as a deformation $\delta = (\delta_x, \delta_y, \delta_w, \delta_h, \delta_\theta)$ of a reference grasp according to the following equation:

$$\begin{aligned} x &= \delta_x * g_{aw} + g_{ax} \\ y &= \delta_y * g_{ah} + g_{ay} \\ w &= \exp(\delta_w) * g_{aw} \\ h &= \exp(\delta_h) * g_{ah} \\ \theta &= \delta_\theta * (180/k) + g_{a\theta} \end{aligned} \tag{2}$$

where k is the number of different anchor boxes. Fig. 2 (b) presents three different examples of oriented anchor boxes.

IV. PROPOSED ARCHITECTURE

A. Network Architecture

Fig. 3 presents the global architecture of our network. There are three main components in it: a features extractor, a grasp predictor and our newly introduced scorer

a) Features extraction: For all our experiments, we used a ResNet-50 network [14] for the features extractor as it has been demonstrated to be efficient in various domains. Depending on the desired size of the features map, the last layers of the ResNet are discarded. In our experiments, we used an image input size of 320×320 pixels and took the output of the fourth convolution block. Therefore, the feature maps have sizes of $20 \times 20 \times 1024$.

b) Grasp prediction: The second part of our network is an oriented anchor box based grasp detector. We used the state-of-the-art architecture presented by Zhou *et al.* [9]. Two convolutional layers are trained to predict, for each pixel of the feature map and for each reference anchor box, a classification score and five regression values. The classification score shows the quality of the corresponding oriented reference grasp: a score close to 1 means a high confidence of a good location while a score close to 0 indicates an inadequate position or orientation for a parallel plate gripper.

c) Scorer: The intermediate grasp quality score, predicted by the prediction network, only depends on the features and the anchor box g_a, and not on the final prediction including the predicted regression values g. So a reference grasp on a good location could have a high score despite being a bad prediction once the final grasp g is computed through Eq. 2. Moreover, the score estimation can not be used to improve the regression quality. To deal with these issues, we extended this state-of-the-art network with a third component: a scorer network. Its role is similar to the Grasp Quality CNN used in different versions of Dex-Net [11] [12] [19]. Using a subset of the features and a grasp representation, the scorer network predicts a probability describing the quality of the proposed grasp.

In Fig. 3, we can see the detailed implementation of this scorer network. We kept this network small to avoid adding too much computation cost and memory usage. A 3×3 area from the feature map around the grasp position is sent through a 1×1 convolutional layer with 1024 filters. Its output is then flattened to a vector of size 9216 ($3 \times 3 \times 1024$). The grasp prediction is set as a vector of size $5 \times k$ where every value is set to 0 except the 5 values corresponding to the considered anchor box which are set to the δ output from the grasp prediction part of the network. Keeping a $5 \times k$ dimension vector allows the network to differentiate the k base anchors while not having to transform the δ coordinates with Eq. 2. This $5 \times k$ vector is passed through a 512 neurons fully connected layer and the result is concatenated with the image vector. The result is a 9728 components vector processed by two last fully connected layers, resulting in a graspability score and a nongraspability score. These two scores are then processed by a softmax to get a grasp probability. All the layers of the scorer network (except the last one) are followed by a leaky ReLU with a negative slope set to 0.1.

B. Anchor selection

As all the grasps predictions are generated regarding reference anchors, choosing them correctly is crucial for the performances of the whole network. In [10], Chu *et al.* used 3 different scales and aspect ratios for a total of 9 axis-aligned anchors. However, Zhou *et al.* showed in [9] that the orientation is more important for accuracy than the dimensions of the box. Therefore, we also used for our experiments only one anchor with 6 different orientations.

However, instead of using a ratio of 1:1 with an arbitrary size, we compute before the training process a mean box of all the ground truth grasps from the training dataset. This approach gives us values for h and w of the anchor boxes and has been proven to give good results in object detection [20]. As grasps usually have a larger w than h, using a mean grasp helps by providing the network reference grasps closer to the real ones than when using a 1:1 ratio.

C. Grasp selection

The grasp prediction network is trained using the fast Angle Matching (AM) strategy presented in [9]. However, for the scorer network, ground truth labels are built using the Jaccard Matching (JM). With Jaccard Matching, a grasp is considered as a good one if it is close (both in orientation and in intersection over union) to a ground truth one. As this is more time-consuming than the Angle Matching, not all the 2400 predictions ($20 \times 20 \times 6$ anchors) are evaluated during training. Only a subset of all the predictions are used to back-propagate gradients in the scorer network. In our experiments, we tested two methods to select the grasps.

In the first one, we do not use the score layer of the prediction network: the scores predicted by the scorer are directly used. All the grasps predicted for all the reference anchors are evaluated with the scorer during the forward pass and the T highest scores are compared to the ground truth and used for backpropagation.

With the second approach, we use the score generated by the prediction network: the T anchor boxes with the highest scores are selected and only the corresponding T grasps are evaluated by the scorer network.

D. Loss functions

There are three different loss functions in our architecture. The classification loss functions for the prediction network and the scorer are both softmax cross-entropy loss. They both

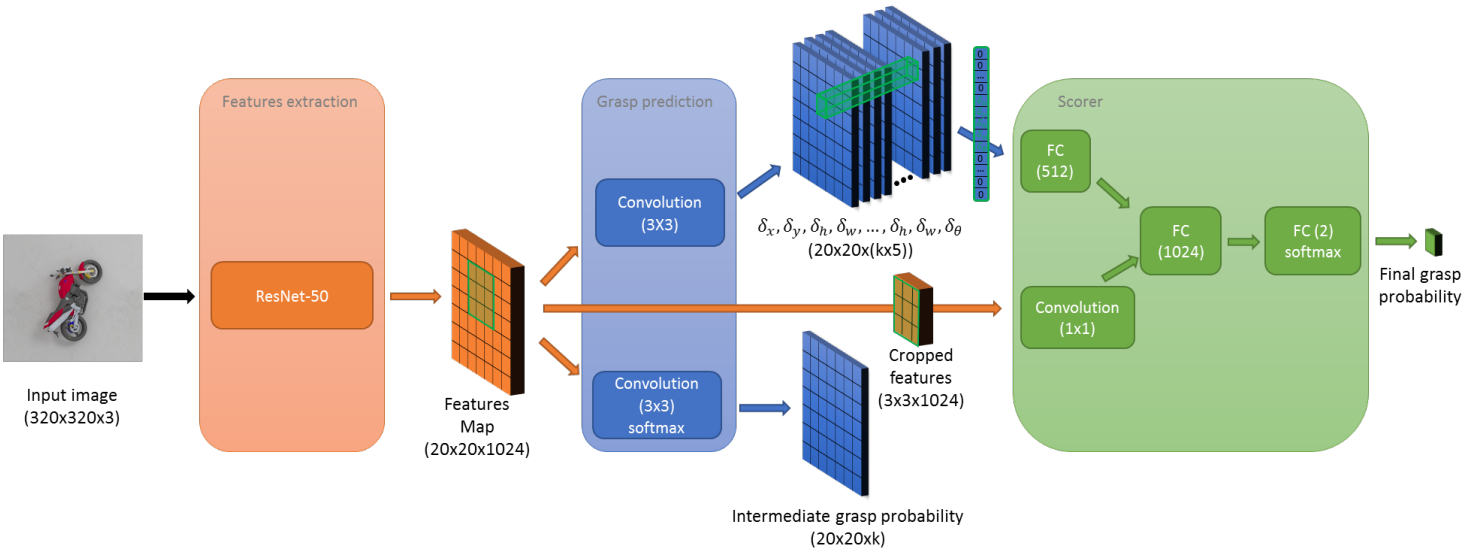

Fig. 3. Global view of our architecture and its three components, the features extractor, the grasp predictor and the scorer network

are computed only on a subset of the predictions: P positive and $3P$ negative for the prediction network one, and T grasps for the scorer one.

$$\mathcal{L}_{cls}(a,p) = -\frac{1}{4P}\sum_{i=1}^{4P} AM(a_i)\log(p_i) + (1 - AM(a_i))\log(1-p_i) \tag{3}$$

$$\mathcal{L}_{scorer}(g,p_s) = -\frac{1}{T}\sum_{i=1}^{T} JM(g_i)\log(p_{si}) + (1 - JM(g_i))\log(1-p_{si}) \tag{4}$$

where a_i is the anchor considered for the i^{th} prediction, g_i is the actual i^{th} predicted grasp, p_i (respectively p_{si}) is the grasp probability outputted by the prediction network (respectively the scorer network) and $AM(a_i)$ (respectively $JM(g_i)$) is 1 if the i^{th} prediction respects the Angle Matching criterion (respectively the Jaccard Matching criterion) and 0 otherwise.

The regression loss is composed of two terms, a classic smooth L1 function to ensure the predicted grasps match the annotated ground truth and a second term involving the scorer network.

$$\mathcal{L}_{reg}(g,p_s) = \frac{\alpha}{P}\sum_{i=1}^{P}\sum_{m\in\{x,y,w,h,\theta\}} L1_{smooth}(\delta_{mi} - \hat{\delta}_{mi}) - \frac{1}{T}\sum_{i=1}^{T}\log(p_{si}) \tag{5}$$

where δ_i are the predicted regression values and $\hat{\delta}_i$ are the ground truth values obtained from the annotated grasp inverting Eq. 2. To be consistent with [9] we kept α to 2 in our trainings. The log part of the loss is an addition with regard to previous work. Through it, the grasp prediction network will be able to use the scorer to estimate the quality of its predictions and modify them to improve their quality. During backpropagation, the gradients from this loss are only used to update the weights of the regression layer and the features extractor, not the scorer network. In the same way, the gradients from $\mathcal{L}_{scorer}$ are not used to update the regression layer but only the scorer network and the features extractor.

V. Jacquard+ Evaluation Dataset

Evaluating and being able to compare models is crucial for research to progress. In order to create an evaluation dataset both challenging and reproducible for all at minimal cost, we used physical simulation. Following the work done in [13], we built an extension of the Jacquard dataset as a reference benchmark for grasp detection algorithm in an environment with reproducible conditions.

The main pipeline for building the dataset is the same. However, to emulate the gap that can exist between real data and training data, we modified some parameters before rendering the image. Instead of having one object seen from a top-view camera, the scenes are composed of several objects and a tilted camera. The angle between the vertical axis and the camera front vector is randomly sampled in the interval $[0, 70°]$. The white texture used for the background is replaced by a one among 30 various textures. Also, lights color, position and intensity are randomized from easy to very difficult conditions. Materials of the objects are generated from a broad range of possibilities, from shiny metal reflecting the light to simple diffuse material. Base color and texture information, when available in the CAD model, are though not modified.

The generated dataset consists in 1000 test images (RGB, depth and masks). Fig. 1 presents some of the images from the test dataset. An interface will be available online to submit grasps predictions and get output performance. We

hope this evaluation dataset will help the research community to build more advanced models that can be used in difficult conditions.

VI. EXPERIMENTS AND RESULTS

A. Dataset used

One of the most common dataset used to train grasp detection networks is the Cornell Grasping Dataset. However, state-of-the-art models achieve very high accuracy rates on this dataset. The few unsuccessful detection results are in fact visually consistent with good grasp locations, even if they do not match any ground truth grasp with Jaccard Matching. So with rates this high, Cornell dataset does not seem to be relevant anymore. Instead, we decided to use the large Jacquard grasping dataset. Just like the Cornell Grasp Dataset, it is composed of RGB-D images with good grasps locations annotated as rectangles, but it has more images (54k instead of 1k) and grasps are more diverse. Like previous work, we divided the dataset in 5 parts and performed cross-validation. This separation was done object-wise which means that images containing one object are either all in the training dataset or all in the testing one. Presented performances are averaged accuracy on the five tests.

As overfitting is a common issue with neural networks, especially with fully connected layers, we used online data augmentation to make sure the network does not see twice the exact same image during the whole training process. In details, the 1024×1024 RGB image is randomly rotated around its center, mirrored, shifted up to 50 pixels in both axis and finally rescaled to 320×320 pixels before being sent into the network. This augmentation is not performed at testing time.

B. Training details

To help the training, all the weights of the features extractor are initialized from a pretraining phase on the large RGB dataset ImageNet [21]. Weights for the other layers are randomly initialized. The network is trained through Stochastic Gradient Descent with a momentum of 0.9 for 100k iterations. As the model is quite memory consuming when not using the intermediate score output, the batch size is only set to 5 for all the models. The learning rate is set to 0.001 and the weight decay to 0.0001. To train the scorer network, T is set to 64 as we found out this was a good trade-off to balance positive and negative examples as well as a value small enough not to increase the training time too much.

C. Experiments

There are two goals to our experiments: showing that (1) the probability predicted by our scorer network is more accurate than the one based only on reference anchor boxes and (2) the gradients introduced by the scorer network improve the quality of the regression generated by the prediction network and the quality of the output of the features extractor. To prove these two points, we trained three different networks on the same data:

- the architecture proposed by Zhou *et al.* in [9] as a baseline reference
- our proposed architecture, without the intermediate score output nor the loss associated to it (Eq. 3). This network uses the first grasp selection mehtod presented in Section IV C.
- our full proposed architecture, with the intermediate score output to select the grasps evaluated by the scorer and used during backpropagation

For our full architecture, there are two possibilities to order the predicted grasps: using the intermediate score or the scorer prediction. We evaluated both of these prediction ordering. As in real life applications the highest ranked grasp is not always the one that is executed by the robot (for example because of physical impossibility for the end-effector to reach the position), we were interested not only in the accuracy regarding the top grasp, but also for the first five and ten grasps.

TABLE I
ACCURACY OF DIFFERENT NETWORKS TRAINED ON JACQUARD DATASET

	Jacquard dataset accuracy		
Architecture	**Top 1 grasp**	**Top 5 grasps**	**Top 10 grasps**
Depierre *et al.* [13]	74.21%	-	-
Zhou *et al.* [9]	81.95%	77.97%	72.07%
Ours, without intermediate score	82.36%	79.73%	75.80%
Ours (intermediate score output)	83.61%	79.40%	73.87%
Ours (scorer output)	**85.74%**	**82.96%**	**79.37%**

Table I presents the detection rates of each network using Jaccard Matching (with standard threshold values of 30° for the angle and 25% for the intersection over union). As can be seen between lines 2 and 3, the accuracy of the scorer network is better than the baseline's, going from 81.95% to 82.36% for the top 1 grasp and from 72.07% to 75.80% for the top 10 grasps. Even if the gap is not large, this shows that the scorer is capable of predicting accurate predictions of the outcome of a grasp, even without the intermediate score used at training time. The second and fourth lines represent the accuracy using the predictions ordered with the intermediate score output, without the scorer at training time for the second line and with it for the fourth. Comparing them allows us to see the influence of the gradients produced by the scorer. The increase from 81.95% without the scorer to 83.61% with it shows that the presence of the scorer at training time helps the network to produce better quality grasps, showing evidences of (2). From the last two lines, we can see that (1) is true: the scores predicted by the scorer are more accurate than those produced by the intermediate score layer as sorting the same grasp predictions using the scorer's predictions lead to a higher accuracy than using the

TABLE II

COMPARISON OF THE ACCURACY FOR DIFFERENT ANGLE THRESHOLD VALUES

Angle Difference	Zhou *et al.* [9]	Ours
30°	81.95%	85.74%
25°	81.76%	85.55%
20°	81.27%	85.01%
15°	80.23%	83.65%
10°	77.79%	80.82%

TABLE III

COMPARISON OF THE ACCURACY FOR DIFFERENT INTERSECTION OVER UNION THRESHOLD VALUES

Intersection over Union	Zhou *et al.* [9]	Ours
20%	85.38%	88.36%
25%	81.95%	85.74%
30%	78.26%	82.58%
35%	74.33%	78.71%

intermediate score output.

To be consistent with the experiments in [9], we also measured accuracy of the networks under different threshold values for the Jaccard index. Table II and Table III show the results for different angle and intersection over union threshold respectively. Both models follow the same pattern, being accurate for angle threshold over 15° and rapidly losing performance after. This might be due to the choice of the orientations and the number of anchors.

D. Jacquard+ evaluation

We performed a Jacquard+ evaluation on both the architecture presented in [9] and our full architecture. To evaluate a predicted grasp, the Simulated Grasp Trial (SGT) criterion introduced in [13] is used in the simulation environment. Fig. 4 shows some examples of the grasps predicted by each of the two networks on images from Jacquard+ dataset. As we can see on the left image, both models predict good grasp location when the conditions are close to the training distribution. However, when the background is more complicated, as on the middle and right images, our model predict more accurate grasps. Overall accuracy on Jacquard+ is 68.8% for Zhou *et al.*'s network and 75.8% for our full architecture.

VII. CONCLUSIONS

In this work, we presented a new architecture combining grasp regression and score evaluation and using the estimation of the score to improve the quality of the regression. We evaluated it on the Jacquard dataset and found it as better to predict grasps than previous state-of-the-art architectures. We also introduced Jacquard+ testing dataset which can be used to evaluate the performance of grasp detection models

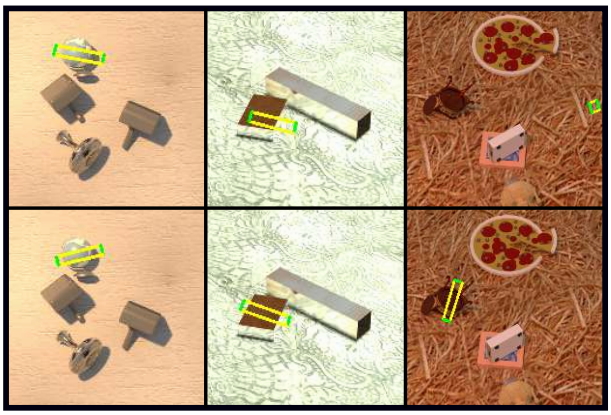

Fig. 4. Comparison of grasp proposals from [9] (top row) and our full architecture (bottom row) on Jacquard+ dataset

in challenging yet reproducible conditions. Our future work will focus on new methods like meta-learning to allow grasp prediction models to quickly adapt to new situations.

REFERENCES

[1] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 1824–1829.

[2] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 362–377, 2010.

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[6] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[7] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1316–1322.

[8] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1609–1614.

[9] X. Zhou, X. Lan, H. Zhang, Z. Tian, Y. Zhang, and N. Zheng, "Fully convolutional grasp detection network with oriented anchor box," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7223–7230.

[10] F.-J. Chu, R. Xu, and P. A. Vela, "Real-world multiobject, multigrasp detection," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3355–3362, 2018.

[11] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," 2017.

[12] J. Mahler, M. Matl, X. Liu, A. Li, D. Gealy, and K. Goldberg, "Dex-net 3.0: Computing robust robot suction grasp targets in point clouds using a new analytic model and deep learning," *arXiv preprint arXiv:1709.06670*, 2017.

[13] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3511–3516.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.
[15] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 769–776.
[16] 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, pp. 91–99.
[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
[19] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1357–1364, 2019.
[20] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.
[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.